

**SYSTEM IDENTIFICATION AND MODEL PREDICTIVE CONTROL  
FOR INTERACTING SERIES PROCESS  
WITH NONLINEAR DYNAMICS**

**TRI CHANDRA SETYO WIBOWO**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING  
UNIVERSITI TEKNOLOGI PETRONAS**

**JULY 2009**

## Status of Thesis

Title of thesis

**System Identification and Model Predictive Control for Interacting Series Process with Nonlinear Dynamics**

I **TRI CHANDRA SETYO WIBOWO** hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐

Confidential

☒

Non-confidential

If this thesis is confidential, please state the reason:

\_\_\_\_\_

The contents of the thesis will remain confidential for \_\_\_\_ – \_\_\_\_ years.

Remarks on disclosure:

\_\_\_\_\_

Endorsed by

**TRI CHANDRA SETYO WIBOWO**

Jl. Ki Ageng Turi  
RT/RW: 19/05  
Kraton, Maospati, Magetan, 63392  
Jawa Timur, Indonesia

Date: \_\_\_\_\_

**ASSOC. PROF. DR. NORDIN SAAD**

Department of Electrical and Electronics  
Engineering  
Universiti Teknologi PETRONAS  
Bandar Seri Iskandar, 31750 Tronoh  
Perak Darul Ridzuan, Malaysia

Date: \_\_\_\_\_

## Approval Page

UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisor (s)

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance, a thesis entitled “**System Identification and Model Predictive Control for Interacting Series Process with Nonlinear Dynamics**” submitted by (**Tri Chandra Setyo Wibowo**) for the fulfillment of the requirements for the degree of Master of Science in Electrical and Electronics Engineering.

\_\_\_\_\_  
Date

Signature : \_\_\_\_\_

Main supervisor : \_\_\_\_\_

Date : \_\_\_\_\_

Signature : \_\_\_\_\_

Co-Supervisor : \_\_\_\_\_

Date : \_\_\_\_\_

UNIVERSITI TEKNOLOGI PETRONAS

System Identification and Model Predictive Control for Interacting Series Process  
with Nonlinear Dynamics

By

Tri Chandra Setyo Wibowo

A THESIS

SUBMITTED TO THE POSTGRADUATE STUDIES PROGRAMME  
AS A REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE  
IN ELECTRICAL AND ELECTRONICS ENGINEERING

BANDAR SERI ISKANDAR

PERAK

JULY 2009

## **Declaration**

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Teknologi PETRONAS (UTP) or other institutions.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

*This thesis is dedicated to my beloved parents, Tarmunadji and Muga Rahayu, who always instill the importance of honesty, humility, hard work, and higher education, and whose words of encouragement and push for tenacity always ring in my ears.*

## **Acknowledgements**

First and foremost I am very grateful towards the Almighty, the Bountiful and the Merciful Allah SWT, with His bestow, I have the strength and opportunity to finish this thesis. Praise be upon Him, with His grace extends my existence to show my gratitude to whom I am going to mention in these following paragraphs.

I would like to express my sincerest gratitude and appreciation to my supervisor, Assoc. Prof. Dr. Nordin Saad, for his continuous assistance, support, and invaluable guidance throughout this research. I would also like to extend my gratitude to my co-supervisor, Assoc. Prof. Dr. Mohd Noh Karsiti, for his guidance during the completion of this research. I also wish to thank Assoc. Prof. Dr. Herman Agustiawan for his assistance, the insightful discussion, and for his lecture on modern control system.

I would like to extend my gratitude to Assoc. Prof. Dr. Rosbi Mamat and Dr. Rosdiazli Ibrahim for their kind efforts in examining my thesis. I would also like to thank Dr. Nor Hisham Hamid for his effort in chairing my presentation.

I would also wish to extend my thanks to the Department of Electrical and Electronics Engineering and the Postgraduate Studies Programme, Universiti Teknologi PETRONAS. I would also like to express my gratitude to all postgraduate office staffs.

An extended acknowledgement to En. Azhar Zainal Abidin for his assistance during my experimental work at the laboratory. I also wish to thank Mr. Totok R. Biyanto, Mr. M. Arrofiq, Mr. Bambang Sumantri, and Mr. Mahidzal Dahari for their kind efforts in helping me during the research.

Tremendous thanks to all my fellow friends, especially Kiki, Pak Gunawan, Erman, Fauzan, Faisal, Aryo, Fikri, Firman Masudi, Surya, Arif, Ayu, Annisa, Amelia, Ariyanti, Iqbal, Ro'il, Teguh, Wawan, Firmansyah, Pak Agni, Pak Ridho, Pak Joko, BKC, Rizki, AW, and Afifi, which always have been very supporting. It helped me keeping the spirit.

Finally, I would like to express my deepest gratitude to my parents, my brothers: Mas Aris and Mas Adhit, my sisters: Mbak Hana, Mbak Yuli, and Dek Yusufa, and my niece, Cahaya Surga, for their patience, open-mindedness, and endless support. They are, always have been and always will be, in my heart.

## **Abstract**

This thesis discusses the empirical modeling using system identification technique and the implementation of a linear model predictive control with focus on interacting series processes. In general, a structure involving a series of systems occurs often in process plants that include processing sequences such as feed heat exchanger, chemical reactor, product cooling, and product separation. The study is carried out by experimental works using the gaseous pilot plant as the process. The gaseous pilot plant exhibits the typical dynamic of an interacting series process, where the strong interaction between upstream and downstream properties occurs in both ways.

The subspace system identification method is used to estimate the linear model parameters. The developed model is designed to be robust against plant nonlinearities. The plant dynamics is first derived from mass and momentum balances of an ideal gas. To provide good estimations, two kinds of input signals are considered, and three methods are taken into account to determine the model order. Two model structures are examined. The model validation is conducted in open-loop and in closed-loop control system.

Real-time implementation of a linear model predictive control is also studied. Rapid prototyping of such controller is developed using the available equipments and software tools. The study includes the tuning of the controller in a heuristic way and the strategy to combine two kinds of control algorithm in the control system.

A simple set of guidelines for tuning the model predictive controller is proposed. Several important issues in the identification process and real-time implementation of model predictive control algorithm are also discussed. The proposed method has been successfully demonstrated on a pilot plant and a number of key results obtained in the development process are presented.



## Abstrak

Thesis ini membahas tentang pemodelan empirikal menggunakan teknik identifikasi sistem dan implementasi satu *model predictive control* yang linier dengan fokus pada proses-proses yang berinteraksi secara bersiri. Secara umum, struktur yang melibatkan rangkaian sistem secara bersiri banyak dijumpai di proses plant yang mempunyai urutan pemrosesan, seperti pada input penukar haba, reaktor kimia, pendinginan produk, dan pemisahan produk. Kajian ini dilakukan dengan eksperimen menggunakan satu *gaseous pilot plant* sebagai proses yang dimodelkan dan dikendalikan. *Gaseous pilot plant* tersebut menunjukkan satu dinamik yang tipikal bagi satu proses yang berinteraksi secara bersiri, di mana berlaku interaksi dominan antara variabel *downstream* dan variabel *upstream*.

Kaedah identifikasi sistem *subspace* digunakan untuk meramalkan parameter-parameter linier dari model. Model yang diusahakan diharapkan dapat mengatasi masalah tidak linear yang ada pada plant. Dinamik plant mula-mula diturunkan dari persamaan jisim dan momentum satu gas ideal. Untuk mendapatkan ramalan yang tepat, dua jenis signal input ditinjau dalam uji plant, dan tiga kaedah dipertimbangkan untuk menentukan *model order*. Dua jenis struktur model juga diuji dalam pemodelan. Pengesahan model dilakukan secara *open-loop* dan juga pada sistem pengendalian *closed-loop*.

Implementasi secara *real-time* satu *model predictive control* yang linier juga dipelajari di thesis ini. *Rapid prototyping* pengendali tersebut dibuat menggunakan peralatan dan perisian komputer yang telah tersedia. Kajian ini melibatkan penalaan (*tuning*) pengendali secara heuristik dan strategi untuk menggabungkan dua jenis algoritma kontrol dalam satu sistem kontrol.

Satu panduan sederhana untuk melakukan penalaan model *predictive control* diusulkan dalam thesis ini. Beberapa masalah penting berhubung proses identifikasi dan implementasi satu *model predictive control* secara *real-time* juga dibincangkan. Kaedah yang diusulkan telah berhasil dibuktikan pada satu pilot plant dan beberapa penemuan penting yang didapati pada proses pengembangan turut dibicarakan.

## List of Publications

The following is a complete list of the publications written during the preliminary work or the work contained in this thesis. This includes accepted and published works.

1. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "The simulation of MISO MPC for gaseous pilot plant control with presence of measurement noise," in *International Conference on Intelligent and Advanced Systems (ICIAS 2007)*, Kuala Lumpur, Malaysia, 2007.
2. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "Multivariable model predictive control with bumpless transfer for gaseous pilot plant control," in *International Conference on Plant Equipment and Reliability (ICPER 2008)*, Kuala Lumpur, Malaysia, 2008.
3. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "Multivariable system identification of gaseous pilot plant," in *National Postgraduate Conference on Engineering, Science, and Technology (NPC 2008)*, Tronoh, Perak, Malaysia, 2008.
4. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "Real-time implementation of embedded model predictive control," in *National Postgraduate Conference on Engineering, Science, and Technology (NPC 2009)*, Tronoh, Perak, Malaysia, 2009.
5. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "System identification of an interacting series process for real-time model predictive control," in *American Control Conference (ACC 2009)*, St. Louis, Missouri, USA, 2009.
6. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "A heuristic approach for tuning model predictive controller," *Jurnal Elektrika*, Submitted.
7. T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "System identification and real-time MPC implementation of an interacting series process," *Journal of Process Control*, Submitted

# Table of Contents

<b>Status of Thesis.....</b>	<b>i</b>
<b>Approval Page .....</b>	<b>ii</b>
<b>Declaration.....</b>	<b>iv</b>
<b>Acknowledgements .....</b>	<b>vi</b>
<b>Abstract.....</b>	<b>vii</b>
<b>Abstrak.....</b>	<b>viii</b>
<b>List of Publications .....</b>	<b>ix</b>
<b>Table of Contents .....</b>	<b>x</b>
<b>List of Figures.....</b>	<b>xiv</b>
<b>List of Tables .....</b>	<b>xviii</b>
<b>List of Abbreviations .....</b>	<b>xx</b>
<b>Nomenclatures.....</b>	<b>xxii</b>
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Motivation and Focus of Present Work .....	1
1.2 Research Objectives.....	2
1.3 Thesis Contributions .....	3
1.4 Thesis Organization .....	5
<b>Chapter 2 Background to Modeling and Control of Interacting Series Process .....</b>	<b>7</b>
2.1 Dynamic Systems Modeling.....	7
2.2 Mass Relationship of Fluids.....	10
2.2.1 Mass balance.....	10
2.2.1.1 <i>Liquid accumulation</i> .....	13
2.2.1.2 <i>Gas accumulation</i> .....	14
2.2.2 Fluid momentum balance.....	15
2.2.3 Flow-through valves .....	16
2.3 Dynamic Behavior of Series Processes.....	17
2.3.1 Noninteracting series process .....	17

2.3.2	Interacting series process .....	19
2.4	Multivariable Control System.....	20
2.5	System Identification .....	22
2.5.1	Overview.....	22
2.5.2	System identification for state-space models.....	25
2.5.3	Advantages and challenges .....	26
2.6	Model Predictive Control.....	30
2.6.1	Historical perspective.....	30
2.6.2	Advantages and challenges .....	32
<b>Chapter 3 System Identification of Interacting Series Process with Nonlinear Dynamics.....</b>		<b>34</b>
3.1	Introduction.....	34
3.2	The Methodology of Modeling and System Identification.....	38
3.3	Subspace Method of System Identification .....	39
3.4	The Gaseous Pilot Plant as an Interacting Series Process.....	41
3.4.1	Plant overview .....	41
3.4.2	Analysis of plant dynamics.....	43
3.4.3	Analysis of plant responses.....	47
3.4.3.1	<i>The responses from changes in the inlet control valve .....</i>	<i>47</i>
3.4.3.2	<i>The responses from changes in the middle control valve .....</i>	<i>49</i>
3.4.3.3	<i>The responses from changes in the outlet control valve .....</i>	<i>50</i>
3.4.4	Models structures.....	51
3.5	System Identification of Gaseous Pilot Plant.....	54
3.5.1	Model 1 .....	55
3.5.1.1	<i>Plant testing .....</i>	<i>55</i>
3.5.1.2	<i>Model structure and variable selection .....</i>	<i>56</i>
3.5.1.3	<i>Order estimation .....</i>	<i>57</i>
3.5.1.4	<i>Identification result and model validation.....</i>	<i>60</i>
3.5.2	Model 2 .....	62
3.5.2.1	<i>Preliminary test.....</i>	<i>64</i>
3.5.2.2	<i>Input signals design .....</i>	<i>65</i>
3.5.2.3	<i>Identification process.....</i>	<i>68</i>
3.5.2.4	<i>Identification result and model validation.....</i>	<i>69</i>

3.5.3	Model 3 .....	71
3.5.3.1	<i>MISO state-space system identification</i> .....	71
3.5.3.2	<i>Combination of MISO models</i> .....	73
3.5.3.3	<i>Identification result and model validation</i> .....	80
3.6	Analysis .....	83
3.7	Summary .....	88
<b>Chapter 4 Real-Time Model Predictive Control of Interacting Series Process with Nonlinear Dynamics.....</b>		<b>89</b>
4.1	Introduction .....	89
4.2	The Methodology of Real-Time Implementation of MPC .....	93
4.3	Model Predictive Control Theory .....	95
4.3.1	Principle of model predictive control .....	95
4.3.2	State estimation .....	97
4.3.3	Prediction of process output .....	99
4.3.4	Unconstrained MPC .....	99
4.3.5	Constrained MPC .....	102
4.3.6	Quadratic programming problem .....	105
4.3.7	Quadratic programming as a control law for constrained MPC .....	106
4.4	Real-Time Controller .....	108
4.4.1	System description .....	109
4.4.2	Generating embedded code and application execution .....	110
4.4.2.1	<i>Simulink simulation steps</i> .....	111
4.4.2.2	<i>Real-time execution</i> .....	112
4.4.3	Changing the parameters .....	114
4.4.4	Data logging and online monitoring .....	115
4.5	Real-Time Implementation of MPC .....	116
4.5.1	Setpoint tracking: Analysis of MPC under model mismatch .....	116
4.5.2	MPC parameters tuning .....	120
4.5.2.1	<i>Controller sensitivities</i> .....	121
4.5.2.2	<i>Horizons</i> .....	125
4.5.2.3	<i>State estimation gain</i> .....	128
4.5.2.4	<i>Reference trajectories</i> .....	132
4.5.3	Discussion .....	136

4.6	MPC and PID .....	139
4.6.1	Tuning of PID .....	140
4.6.2	Control performances.....	143
4.6.3	Switching control and bumpless transfer .....	147
4.6.4	Discussion .....	151
4.7	Summary .....	151
<b>Chapter 5 Conclusions and Future Work .....</b>		<b>153</b>
5.1	Conclusions.....	153
5.2	Directions for Future Works .....	154
<b>References.....</b>		<b>156</b>

## List of Figures

Figure 2.1: Mass balance in a liquid tank with variable volume and fixed pressure.....	11
Figure 2.2: Mass balance in a pressure chamber with variable pressure and fixed volume. .....	11
Figure 2.3: Reference volume in a fluid pipe [38].....	16
Figure 2.4: A pH neutralization process with noninteracting series structure.....	18
Figure 2.5: The transfer function of noninteracting series process.....	18
Figure 2.6: A process with interacting series structure.....	19
Figure 2.7: A system with input, output, and disturbance. ....	23
Figure 2.8: A flow diagram of system identification [43, 44]. ....	25
Figure 2.9: Subspace and classical methods of state-space system identification [43, 45]. .....	26
Figure 3.1: The flow diagram of plant modeling and system identification.....	38
Figure 3.2: Schematic diagram of the gaseous pilot plant. ....	42
Figure 3.3: Photograph of the gaseous pilot plant. ....	43
Figure 3.4: The step changes in inlet control valve (top), the upstream response (left- bottom), and the downstream response (right-bottom) of the gaseous pilot plant, while the two others control valve are kept to be constant at 50% of valve openings. ....	47
Figure 3.5: The step changes in middle control valve (top), the upstream response (left- bottom), and the downstream response (right-bottom) of the gaseous pilot plant, while the two others control valve are kept to be constant at 50% of valve openings. ....	49
Figure 3.6: The step changes in outlet control valve (top), the upstream response (left- bottom), and the downstream response (right-bottom) of the gaseous pilot plant, while the two others control valve are kept to be constant at 50% of valve openings. ....	50
Figure 3.7: The dynamics of gaseous pilot plant. ....	51
Figure 3.8: Input signals and plant responses for the development of Model 1. ....	56
Figure 3.9: The schematic diagram of the first model structure. ....	57

Figure 3.10: Model singular values plot with the suggested model order (red) given by N4SID algorithm.....	57
Figure 3.11: Hankel singular values plot. ....	58
Figure 3.12: Relative square error (RSE) plot of Model 1. ....	59
Figure 3.13: Location of the poles of Model 1. ....	61
Figure 3.14: The comparison of real output and simulated output generated by Model 1 for upstream pressure (top) and downstream pressure (bottom). ....	62
Figure 3.15: PRBS signals and plant responses for the development of Model 2.....	68
Figure 3.16: Location of the poles of Model 2. ....	70
Figure 3.17: The comparison of real output and simulated output generated by Model 2 for upstream pressure (top) and downstream pressure (bottom). ....	70
Figure 3.18: The schematic diagram of the second model structure which consists of two MISO models. ....	71
Figure 3.19: Input signals and plant responses for the development of Model 3.....	72
Figure 3.20: Location of the poles of Model 3. ....	82
Figure 3.21: The comparison of real measured and simulated outputs generated by Model 3 for upstream pressure (top) and downstream pressure (bottom). ....	83
Figure 3.22: APRBS input of the three control valves for model validation.....	84
Figure 3.23: The comparison of the real output from APRBS response data and the simulated outputs generated by Model 1, Model 2, and Model 3 for the upstream pressure. ....	85
Figure 3.24: The comparison of the real output from APRBS response data and the simulated outputs generated by Model 1, Model 2, and Model 3 for the downstream pressure. ....	86
Figure 4.1: The flow diagram of real-time model predictive control implementation. ....	94
Figure 4.2: Model predictive control illustration.....	95
Figure 4.3: Geometric of 2-dimensions quadratic programming illustration. ....	107
Figure 4.4: The real-time control system architecture. ....	109



Figure 4.5: Photograph of the xPC TargetBox with the I/O module (left) and the host PC (right). .....	110
Figure 4.6: Steps in a Simulink simulation. ....	111
Figure 4.7: Simulink diagram for offline simulation of MPC. ....	111
Figure 4.8: Real-time execution of the model code. ....	113
Figure 4.9: Simulink diagram that ready to be downloaded into xPC TargetBox for real-time MPC. ....	114
Figure 4.10: The schematic diagram of the closed-loop MPC control system. ....	116
Figure 4.11: Upstream responses of MPC control system with variation of the internal model. ....	118
Figure 4.12: Downstream responses of MPC control system with variation of the internal model. ....	119
Figure 4.13: Upstream responses of MPC control system with variation of the weights of objective function. ....	123
Figure 4.14: Downstream responses of MPC control system with variation of the weights of objective function. ....	124
Figure 4.15: Upstream responses of MPC control system with variation of prediction and control horizons. ....	126
Figure 4.16: Downstream responses of MPC control system with variation of prediction and control horizons. ....	127
Figure 4.17: Upstream responses of MPC control system with variation of the state estimation gain. ....	130
Figure 4.18: Downstream responses of MPC control system with variation of the state estimation gain. ....	131
Figure 4.19: Upstream responses of MPC control system with and without applying the reference trajectory filter. ....	133
Figure 4.20: Downstream responses of MPC control system with and without applying the reference trajectory filter. ....	134

Figure 4.21: Upstream responses of MPC control system with variation of the reference trajectory filter. ....	135
Figure 4.22: Downstream responses of MPC control system with variation of the reference trajectory filter.....	136
Figure 4.23: Sustain oscillation of the upstream process variable (PT202). ....	141
Figure 4.24: Sustain oscillation of the downstream process variable (PT212).....	142
Figure 4.25: The comparison of MPC and PID in the upstream control system as the response of setpoint change. ....	143
Figure 4.26: The comparison of MPC and PID in the downstream control system as their responses to setpoint change. ....	144
Figure 4.27: The comparison of manipulated variable movements in the upstream control system using PID and MPC. ....	145
Figure 4.28: The comparison of manipulated variable movements in the upstream control system using PID and MPC. ....	145
Figure 4.29: The spectrum of manipulated variable in steady-state condition given by PID and MPC controllers for the upstream control system.....	146
Figure 4.30: The spectrum of manipulated variable in steady-state condition given by PID and MPC controllers for the downstream control system.....	147
Figure 4.31: The schematic diagram of PID and MPC in the control system. ....	148
Figure 4.32: The manipulated variable (top) and the process variable (bottom) of upstream control system during the switching time. ....	150
Figure 4.33: The manipulated variable (top) and the process variable (bottom) of downstream control system during the switching time.....	150

## List of Tables

Table 3.1: The Performance of Model 1.....	62
Table 3.2: The FOPDT transfer functions of plant outputs from the respective inputs.....	65
Table 3.3: The standard statistical properties of signals and noises .....	65
Table 3.4: PRBS signals design of the three plant inputs .....	67
Table 3.5: The performance of Model 2. ....	71
Table 3.6: The performance of Model 3 .....	82
Table 3.7: Performance comparison of Model 1, Model 2, and Model 3.....	86
Table 4.1: MPC controllers performance parameters with variation of internal model ..	120
Table 4.2: MPC controllers performance parameters with variation of the weights of objective function for the upstream responses.....	123
Table 4.3: MPC controllers performance parameters with variation of the weights of objective function for the downstream responses.....	124
Table 4.4: MPC controllers performance parameters with variation of prediction and control horizons for the upstream responses.....	127
Table 4.5: MPC controllers performance parameters with variation of prediction and control horizons for the downstream responses.....	128
Table 4.6: MPC controllers performance parameters with variation of the state estimation gain for the upstream responses.....	130
Table 4.7: MPC controllers performance parameters with variation of the state estimation gain for the downstream responses .....	131
Table 4.8: MPC controllers performance parameters with and without applying the reference trajectories filter .....	134
Table 4.9: MPC controllers performance parameters with different time constant of the reference trajectories .....	136
Table 4.10: Ziegler-Nichols closed-loop tuning correlations .....	141

Table 4.11: Controllers tuning parameters based on the Ziegler-Nichols closed-loop tuning method .....	142
Table 4.12: Controllers performance parameters of PID and MPC controllers .....	143

## **List of Abbreviations**

A/D	Analog-to-Digital
APC	Advanced Process Control
APRBS	Amplitude-Modulation Pseudo-Random Binary Sequence
ARMAX	Auto Regressive Moving Average with eXogenous
ARX	Auto Regressive with eXogenous
BIOS	Basic Input/Output System
CACSD	Computer Aided Control System Design
CAN	Controller Area Network
CPC	Chemical Process Control
D/A	Digital-to-Analog
DMC	Dynamic Matrix Control
FCV	Flow Control Valve
FIR	Finite Impulse Response
FOPDT	First-Order Plus Delay Time
I/O	Input/Output
IDCOM	Identification and Command
IMC	Internal Model Control
LSB	Least Significant Bit
LTI	Linear Time-Invariant
MAC	Model Algorithmic Control
MIMO	Multiple-Inputs Multiple-Outputs
MISO	Multiple-Inputs Single-Output
MPC	Model Predictive Control
N4SID	Numerical algorithm for Subspace State-Space System Identification
NLMPC	Non-Linear Model Predictive Control
OE	Output Error
PCV	Pressure Control Valve
PEM	Prediction Error Method
PID	Proportional Integral Derivative
PRBS	Pseudo-Random Binary Sequence

PT	Pressure Transmitter
PWM	Pulse-Width Modulation
QP	Quadratic Programming
QR decomposition	A decomposition of the matrix into an orthogonal matrix Q and a right triangular matrix R
RMPCT	Robust Multivariable Predictive Control Technology
RPN	Robust Performance Number
RSE	Relative Square Error
RTW	Real-Time Workshop
SMOC	Shell Multivariable Optimizing Controller
UTP	Universiti Teknologi PETRONAS

## Nomenclatures

### Latin symbols

<b>A</b>	State coefficient matrix $\mathbf{A}_i$ = specific state coefficient matrix shown in subscript
<b>a</b>	Any nonzero complex vector
<b>A</b>	Cross-sectional area of pipe
<b>a</b>	Amplitude of signal
$a_{l-ij}$	Element of $i^{\text{th}}$ -row and $j^{\text{th}}$ -column of a specific state coefficient matrix denoted in $l$
<b>B</b>	Input coefficient matrix $\mathbf{B}_i$ = specific input coefficient matrix shown in subscript
$b_{l-ij}$	Element of $i^{\text{th}}$ -row and $j^{\text{th}}$ -column of a specific input coefficient matrix denoted in $l$
<b>C</b>	Output coefficient matrix $\mathbf{C}_i$ = specific output coefficient matrix shown in subscript
<b>c</b>	The real vector formed in equation (4.59)
$Cap_{m,gas}$	Mass capacity of the gas at constant temperature in a fixed volume
$Cap_{v,gas}$	Compression capacity of the gas at a certain temperature and fixed volume
$C_i$	Specific concentration of unit shown in subscript
$C_v$	Valve coefficient
<b>c</b>	Integer
$c_{l-ij}$	Element of $i^{\text{th}}$ -row and $j^{\text{th}}$ -column of a specific output coefficient matrix denoted in $l$
$c_p$	Specific heat at constant pressure
$c_v$	Specific heat at constant volume

<b>D</b>	Feedthrough coefficient matrix $\mathbf{D}_i$ = specific feedthrough coefficient matrix shown in subscript
$\mathcal{D}$	Data set obtained by input-output measurements
$d_{l-ij}$	Element of $i^{\text{th}}$ -row and $j^{\text{th}}$ -column of a specific feedthrough coefficient matrix denoted in $l$
<b>F</b>	The real matrix formed in equation (4.43)
$\mathbf{F}_u$	The real matrix formed in equation (4.17)
$\mathbf{F}_x$	The real matrix formed in equation (4.16)
<b>f</b>	Vector of free responses of a system
$\mathbf{f}_1$	The real vector formed in equation (4.35)
$\mathbf{f}_2$	The real vector formed in equation (4.43)
$\mathbf{f}_c$	The real vector formed in equation (4.60)
<b>F</b>	Flow $F_m$ = mass flow $F_{m,i}$ = specific mass flow shown in subscript $F_{m,in}$ = inlet mass flow $F_{m,in,i}$ = specific inlet mass flow shown in subscript $F_{m,out}$ = outlet mass flow $F_{m,out,j}$ = specific outlet mass flow shown in subscript $F_v$ = volumetric flow $F_{v,i}$ = specific volumetric flow shown in subscript $F_{v,in}$ = inlet volumetric flow $F_{v,in,i}$ = specific inlet volumetric flow shown in subscript $F_{v,out}$ = outlet volumetric flow $F_{v,out,j}$ = specific outlet volumetric flow shown in subscript



	Force, equations (2.26) to (2.28)
	$F_{external}$ = external force, equations (2.26) to (2.28)
$f(u)$	Fraction of the total flow area of the valve
<b>G</b>	The real matrix formed in equation (4.18)
<b>g</b>	First derivative of MPC objective function, equation (4.25)
<b>g'</b>	Second derivative of MPC objective function, equation (4.26)
<b>G</b>	Transfer function
	$G_i$ = specific transfer function shown in subscript
	$G_{ij}$ = transfer function of $i^{th}$ -input to $j^{th}$ -output, equations (3.29) to (3.39)
	$G_{y-ij}$ = transfer function of $i^{th}$ -output to $j^{th}$ -output, equations (3.29) to (3.39)
$g(i\omega)$	Frequency response
<b>H</b>	A positive definite matrix, equation (4.27)
	The Hessian matrix of quadratic programming MPC problem, equations (4.47) to (4.61)
$h$	Integer
<b>I</b>	The identity matrix
$i$	Imaginer
	Integer
<b>J</b>	MPC objective function
$j$	Integer
<b>K</b>	State estimation gain
$\tilde{K}$	The modified state estimation gain, equation (4.63)
<b>K</b>	Process gain
	$K_i$ = specific process gain shown in subscript
	Constant coefficient, equations (3.10) to (3.26)
	$K_i$ = specific constant coefficient shown in subscript, equations (3.10) to (3.26)

$K_{ss}$	constant coefficient during steady state, equations (3.10) to (3.26)
$K_{ss,i}$	specific constant coefficient during steady state shown in subscript, equations (3.10) to (3.26)
$K_c$	PID controller gain
$K_u$	Ultimate gain of PID controller using Ziegler-Nichols closed-loop tuning method
$k$	Integer
$\mathcal{L}$	Lagrangian function
$l$	Length of pipe Integer
$\mathcal{M}$	Model set or model structure, containing candidate models
$M$	Molecular weight
$m$	Mass Integer Length of control horizon
$N$	Number of data
$n$	Number of moles, equation (2.18) System order Number of shift registers, sub-sub-section 4.4.2.2 Length of minimum horizon
$n_u$	Size of input vector
$n_x$	Size of state vector
$n_y$	Size of output vector
$\mathbf{P}$	Controllability grammian, equations (3.40) and (3.41) Solution of an algebraic Ricatti equation, equations (4.13 and 4.14)

$P$	Pressure $P_i$ = specific pressure shown in subscript $P_v$ = pressure across the valve $P_s$ = source pressure $P_{amb}$ = ambient pressure Period, equations (3.49) to (3.56)
$P_u$	Ultimate period of PID controller using Ziegler-Nichols closed-loop tuning method
$p$	Length of prediction horizon
$Q$	Covariance matrix of state residual, equation (3.5) Observability grammian, equations (3.40) and (3.42) Output weight matrix
$Q$	Diagonal element of output weight matrix
$R$	Covariance matrix of noise $R_w$ = covariance matrix reflecting a plant noise level $R_v$ = covariance matrix reflecting a measurement noise level Input weight matrix
$r$	Reference trajectory
$\mathbb{R}$	Real number domain
$R$	Gas constant Diagonal element of input weight matrix
$S$	Covariance matrix of state and noise residual, equation (3.5)
$s.g.$	Specific gravity of fluid
$T$	Temperature $T_i$ = specific temperature shown in subscript Number of clock ticks in every period, equations (3.49) to (3.56)
$T_c$	Clock tick time

$T_d$	Derivative time of PID controller
$T_I$	Integral time of PID controller
$t$	Time
$\mathbf{u}$	Input vector $\mathbf{u}_i$ = specific input vector shown in subscript $\mathbf{u}_c$ = common input vector $\mathbf{u}_{ind}$ = individual input vector $\mathbf{u}_{ind-i}$ = specific individual input vector denoted in $i$ $\mathbf{u}_{min}$ = minimum constraint of input vector $\mathbf{u}_{max}$ = maximum constraint of input vector
$u$	Input of process or system $u(t)$ = input of process or system in continuous-time domain $u(k)$ = input of process or system in discrete-time domain $u_i$ = specific input shown in subscript $u(i\omega)$ = Fourier transforms of input Fraction of valve opening, equations (2.32) to (2.35)
$\mathbf{v}$	Vector of measurement noise
$V$	Volume
$V_N$	Criterion to select the best model(s)
$v$	Velocity
$\mathbf{w}$	Vector of plant noise
$\mathbf{X}$	State sequence matrix
$\mathbf{x}$	State vector $\hat{\mathbf{x}}$ = estimated state vector $\mathbf{x}_i$ = specific state vector shown in subscript
$X$	Inlet concentration, equations (2.36) and (2.37)
$x$	Composition, equation (2.8)

	$x_0$ = composition in the operating point, equation (2.8)
$x_{l-i}$	Element of $i^{\text{th}}$ -row of a specific state vector denoted in $l$
$\mathbf{y}$	Output vector $\mathbf{y}_i$ = specific output vector shown in subscript $\hat{\mathbf{y}}$ = predicted output vector $\hat{\mathbf{y}}_{\min}$ = minimum constraint of predicted output vector $\hat{\mathbf{y}}_{\max}$ = maximum constraint of predicted output vector
$Y$	Outlet concentration, equations (2.36) and (2.37) $Y_i$ = specific outlet concentration shown in subscript, equations (2.36) and (2.37)
$y$	Output of process or system $y(t)$ = output of process or system in continuous-time domain $y(k)$ = output of process or system in discrete-time domain $y_i$ = specific output shown in subscript $\hat{y}$ = predicted output $\hat{y}_i$ = specific predicted output shown in subscript $\bar{y}$ = mean of output $y(i\omega)$ = Fourier transform of output

### Greek symbols

$\alpha$	Constant factor
$\beta$	Isothermal bulk modulus Constant factor, equations (3.50) and (3.54)
$\beta_{ab}$	Adiabatic bulk modulus
$\chi$	The constraint vector of inequality constraint of general quadratic programming problem, equations (4.49) to (4.55)
$\varepsilon$	Vector of prediction error

$\hat{\mathbf{e}}$	Vector of closed-loop prediction error based on future and current control actions
$\Phi$	The Hessian matrix of general quadratic programming problem, equations (4.49) and (4.50)
$\Phi_{uu}(\omega)$	Power spectrum of signal
$\Gamma$	Extended observability matrix
$\gamma$	Thermal expansion coefficient
$\eta$	Vector of state residual
$\kappa$	Vector of slack variables of general quadratic programming problem, equations (4.51) to (4.55)
$\Lambda$	Noise coefficient matrix
$\lambda$	Lagrange multiplier for inequality constraint of general quadratic programming problem, equations (4.50) to (4.55)
$\lambda$	Eigenvalue of $\mathbf{PQ}$ , equation (3.40)
$\pi$	Lagrange multiplier for variable bound of general quadratic programming problem, equations (4.50) to (4.55)
$\Theta$	System parameters matrix, equation (3.3)
$\theta$	The vector variable of general quadratic programming problem, equations (4.49) to (4.55)
$\rho$	Density $\rho_i$ = specific density shown in subscript
$\sigma$	Standard deviation $\sigma_s$ = standard deviation of signal $\sigma_N$ = standard deviation of noise
$\sigma_H$	Hankel singular value
$\tau$	Time constant $\tau_i$ = specific time constant shown in subscript

$\Upsilon$	Set of model parameters
$\mathbf{v}$	Vector of output residual
$\nu$	Model parameters
$\Omega$	The variable coefficient matrix of inequality constraint of general quadratic programming problem, equations (4.49) to (4.55)
$\omega$	Frequency of signal $\omega_{low}$ = lowest frequency present in signal $\omega_{high}$ = highest frequency present in signal $\omega_{PRBS}$ = frequencies contained in PRBS signal $\omega_{min}$ = minimum frequency of PRBS signal $\omega_{max}$ = maximum frequency of PRBS signal
$\Xi$	The real matrix formed in equation (4.35)
$\xi$	The vector coefficient of general quadratic programming problem, equations (4.49) and (4.50)
$\Psi$	The input coefficient matrix in inequality constraint (4.47)
$\Psi$	The constraint vector in inequality constraint (4.48)

# **Chapter 1**

## **Introduction**

### **1.1 Motivation and Focus of Present Work**

As process technology grows rapidly, the degree of competition in all branches of the process industry increases. This situation puts pressure on each single site and plant to stay competitive. Even within a company there is an internal competition of being the most productive and effective, and delivering the best quality products [1]. Optimal economic operation of processes is one of the most important aspects to achieve or maintain the competitive advantages in process industry [2]. Plant operation depends on its control structure and plantwide control related to that design for complete process plants [3].

To increase the efficiency in process plants, their processing units tend to become more tightly integrated or interconnected, e.g., the raw materials are recycled and the hot process steams are heat exchanged with the cold process steams, so that the resources can be used more economically [4]. However, tight integration may make the plants more difficult to operate and control. Morud [4] points out that as integration become tighter, the behavior of the plant is modified, and may become very different from what could be expected by looking at the individual processing units. Furthermore, in some cases, integration may lead to instability, poor controllability, introducing new steady-states, limit cycles, or even chaotic behavior [4].

The advances of computer-based control system have brought an incredible revolution in process industries, especially in their control system strategy. Without any doubt, model predictive control (MPC) has attracted notable attention in process industries and become the most popular advanced process control (APC) strategy over past few years [5-13]. MPC consists of step-by-step optimization technique and offers the possibility to handle constraints on input and output signals during the design and implementation of the controller [14]. From the control engineering viewpoint, MPC promises a great benefit to maintain the optimal economic operation of plant and preserve the lifetime of equipments. One of the main drawbacks of MPC is the difficulty to incorporate model



uncertainties of plant explicitly, and for this reason, increasing attention has been placed on robust MPC problems [13-17].

In practice, the implementation procedure of MPC can be divided into two stages. The first is on developing the plant model, and the second is on developing the controller. Mostly, the system identification techniques are preferred to perform the plant modeling. Deriving the plant model based only on the physical laws for control purposes have rarely been attempted in practice. For a plant which consisting of interconnection structure, the problems of developing the plant model using the system identification techniques needs to be addressed. In particular, a number of issues regarding this problem arise such as how to perform the plant testing [18, 19] and what type of model structure should be considered [18-22]. A number of issues also arise in the real-time implementation of model predictive controller, ranging from how to design the hardware for the real-time implementation of MPC [23-25], to what extend do the model uncertainties effect the controller robustness [13-17], how to tune the controller [26-29], and how to design and evaluate a nonlinear MPC [30-34].

These issues, followed by the eager for a better understanding of the theoretical insight as well as actual practice of MPC, are motivated this work in which the aim is to continue the investigation on the practical design and implementation of model predictive controller.

## 1.2 Research Objectives

The objectives of this research are stated as follows:

- a. To study and investigate the empirical modeling of an interacting series process using system identification techniques.
- b. To study and perform the real-time implementation of MPC algorithm for controlling an interacting series process.

The scope of this research is limited to the following:

- a. The investigation of the interacting series process being modeled and controlled is performed using the gaseous pilot plant as a case study.
- b. Only linear time-invariant (LTI) models and linear controllers are considered.

### 1.3 Thesis Contributions

The main contributions of this thesis are:

- a. The idea of using multiple-input single-output (MISO) structures in the system identification processes and the formulation to construct a multiple-input multiple-output (MIMO) state-space model for developing the empirical model of an interacting series process.

With the development of subspace system identification method, which is designed to be able to perform MIMO identification, several researchers see that the identification of multivariable processes using MISO structures is not efficient, in view that there are common or correlated parameters among models for different output variables and/or correlated noise [18, 20]. However, it is found in this research that the LTI models obtained from MIMO identification are not robust against the plant nonlinearities. The robustness is shown when the LTI model constructed from MISO identification is used. The formulation to construct a MIMO state-space model from several MISO models is also presented in this thesis.

- b. A simple set of guidelines for tuning MPC controller.

Based on the results of heuristic approach for tuning a linear model predictive control in real-time conditions, a simple set of guidelines for tuning MPC controller is proposed in this thesis. The guidelines consist of simple rules to adjust the parameters used to tune the MPC controller.

The gaseous pilot plant is one of the available equipments for research purposes in Process Instrumentation and Control Laboratory, Department of Electrical and Electronics Engineering, Universiti Teknologi PETRONAS (UTP) Malaysia. Currently, there are several undergraduate and postgraduate students doing their research or project based on this plant. Therefore, the outcomes of the researches related with this plant would be beneficial for the current and future researches in the area of Process Instrumentation and Control. The contributions of this thesis associated with such purposes include the following:

- a. The dynamics of the gaseous pilot plant based on the mass and momentum balances of an ideal gas.

Before determining the structure and estimating the parameters of the plant model, the dynamics of the gaseous pilot plant is first derived based on the mass and momentum balances of an ideal gas. This derivation employs several linearizations. Should there be no retrofitting of the gaseous pilot plant, the result may also be used in further analysis and design of control and monitoring system in such a plant.

- b. The comparison of MIMO identification using step signals with sequential excitation and pseudo-random binary sequence (PRBS) signals with simultaneous excitation.

Several researchers, for example Li and Georgakis [35] and Micchi and Pannochia [36], point out that for MIMO identification, the identified system can be perturbed simultaneously using PRBS signal. However, based on this research, that setting may not work well for certain plants. In some cases, the sequential excitation may be preferred. Such a comparison is presented in this thesis, and it is expected that the finding may serve as an insight for further investigation in the empirical modeling of the gaseous pilot plant using system identification techniques, especially in the design of the input signals.

- c. Rapid prototyping of an embedded model predictive controller using off-the-shelf components.

The MPC algorithm presented in this thesis is carried out in real-time using available equipments, components, and software. Several available features have also been tested, and it is found that the gaseous pilot plant can indeed be controlled using such a prototyping environment. Hence, it is also plausible to perform further investigation in the implementation of MPC algorithm without necessarily have to write the low-level code or design a compatible data acquisition card.

- d. Steady-state bumpless transfer of MPC using external feedback of manipulated variables.

Bemporad et al. [37] present a method to make a smooth transition when the MPC controller is switched from manual to automatic mode. In this thesis, such a method is demonstrated in the real-time control of the gaseous pilot plant to produce the smooth transition (bumpless transfer) of actuators when the controller is switched from

proportional-integral-derivative (PID) to MPC in the steady-state condition. Hence, should there be a further investigation of the implementation of MPC in the gaseous pilot plant, the use of the external feedback of manipulated variables to make a steady-state bumpless transfer may also worthwhile to be considered.

## 1.4 Thesis Organization

This thesis is composed of six chapters with a joint bibliography. This chapter (Chapter 1) presents the motivation for the research, objective, and scope of study. The contributions of this thesis are also highlighted.

Chapter 2 reviews some of the research work as the background of present research. The chapter starts with a discussion on the modeling of the dynamic systems including the classification of models used in most applications. The review is continued with outlines of the mass relationship of fluid, the dynamic behavior of series processes, and the idea of multivariable control system. A brief overview of system identification and model predictive control including their advantages and challenges are also presented.

Chapter 3 presents the development of the empirical model of an interacting series process using system identification technique. The review of the related works is first presented in the introduction section. The methodology of the model development and analysis is presented in the subsequent section. Then, details of the identification processes are reported. Several important issues in the development of the empirical model of the interacting series processes are also discussed. A summary is given in the last section of this chapter.

Chapter 4 presents the implementation of MPC algorithm in real-time control system. In the introduction, several related researches are presented. The methodology of the research for this particular part is presented in the subsequent section. The formulation of model predictive control algorithm is outlined in the second section. Details of the work are reported in the next three sections, started from design and configuration of a real-time MPC controller, real-time implementation of model predictive control including analyses of model mismatch and controller tuning, and also the study of an alternative approach to combine PID and MPC algorithms in the controller. A summary is also given in the last section of this chapter.

Finally, Chapter 5 gives the conclusions of this thesis and proposes future research directions relevant to system identification and model predictive control especially for the interacting series processes.

## **Chapter 2**

### **Background to Modeling and Control of Interacting Series Process**

This chapter presents the review of the recent literatures and researches related to the present work. The review includes the overview of the dynamic system modeling and model classification, the basic theory of the mass relationship of fluid, the dynamic behavior of series processes, and the idea of multivariable control system. In the last two sections of this chapter, several advantages and current challenges in the system identification and model predictive control are outlined.

#### **2.1 Dynamic Systems Modeling**

Modeling is a procedure to formulate the dynamics effects of a system that will be considered into mathematical equations. The dynamic behavior can be characterized by the dynamic responses of the system to the inputs and disturbances, taking into account the initial conditions of the system. The different applications of models and the different modeling goals have lead to many different model structures. The model should be presented in a usable form, so that it should not be too complex, nevertheless it should give a sufficiently accurate description of the system. According to Roffel and Betlem, the model can be classified as follows [38]:

a. **White box versus black box models.**

White box models are based on physical and chemical laws of conservation, such as mass balance, component balance, momentum balance, and energy balance. The models give physical insight into the process and explain the process behavior in terms of state variables and measured variables. The state variable of the model is the variable whose rate of change is described by the conservation balance. These models can already be developed when the process does not yet exist. White box modeling is also known as physical modeling.

Black box models or empirical models do not describe the physical phenomena of the process. They are based on input-output data and only describe the relationship between the measured input and output data of the process. These models are useful

when limited time is available for model development and/or when there is insufficient physical understanding of the process.

It could be that much physical insight is available, but that certain information or understanding is lacking. In those cases, physical models could be combined with black box models. The resulting models are called gray box or hybrid models.

b. Parametric versus non-parametric models.

Parametric models consist of a set of equation that express a set of quantities as explicit functions of several independent variables, known as “parameters”. Parametric model need exact information about the inner structure and have a limited number of parameters.

c. Linear versus non-linear models.

Linear models have a property that is called superposition. That is, if for a given input,  $u_1(t)$ , the response is  $y_1(t)$ , and for an input,  $u_2(t)$ , the response is  $y_2(t)$ , then for an input  $u(t) = \alpha_1 u_1(t) + \alpha_2 u_2(t)$  the response is  $y(t) = \alpha_1 y_1(t) + \alpha_2 y_2(t)$ . The response of a linear model is also proportional to the input. That is, if for a given input,  $u(t)$ , the response is  $y(t)$ , then for an input  $\alpha u(t)$  the response is  $\alpha y(t)$ ,  $\alpha \in \mathbb{R}$ .

Linear dynamic models, such as time series models, are often used as an approximation of the true relationship between the process input and process output. Linear models often provide an accurate description of reality provided the operating range is limited. In that case, a first-order Taylor series approximation is appropriate, where the first-order derivative is used to describe the behavior around the operating point. Linear models are much easier to handle mathematically and easier to interpret as the relationship between input and output is explicit.

It is generally not easy to solve non-linear models analytically, but sophisticated numerical methods embedded in commercial software packages are available to deal with several classes of non-linear models. Developing a non-linear model requires much insight and understanding of the developer as to what mechanism underlies the observed data.

d. Static versus dynamic models.

In static model, the most recent output or dependent variables depend on the most recent values of the input or independent variables. In dynamic models, the state variables describe the change in dependent variable as a function of independent variables and time, the response is called the system transient.

In process engineering, static model are, among others, used in optimization and process design, whereas dynamic models find their application in process control and prediction of the values of future process variables.

e. Distributed versus lumped parameter models.

In some cases, the independent and dependent process variables can vary along a spatial coordinate. For instance in a tubular reactor, temperatures, concentrations, and other process variables vary with the axial coordinate of the reactor. In the absence of micro-mixing, the system variables vary with the radial coordinate of the reactor. Spatial variations of process variables lead to complex models and consequently the solution of the model is difficult. Models that account for spatial variations are called distributed parameter models. A widely used method that approximates the behavior of distributed parameter systems is the division of the spatial coordinate into small sections, within each section the system properties are assumed to be constant. Each section can then be considered as an ideally mixed section and the entire process is approximated by a series of ideally mixed sub-systems. Such a system approximation is called a lumped parameter system.

The advantage of a lumped parameter model over a distributed parameter model is that the lumped parameter model is much easier to solve. However, it needs to be ensured that the lumped parameter representation is an adequate approximation of the true process behavior. So that, some adequacy criteria of the model will be very useful.

f. Time domain versus frequency domain models.

For basic process control purpose, the goal is to eliminate or to compensate disturbances. The influence of these disturbances is determined by the amplitude and the frequency of the changes. Therefore, control engineers also like to describe systems in the frequency or Laplace domain.



The frequency domain offers the possibility to compose in an easy way the overall input-output relation of a complex system consisting of several interconnected subsystems, from the separate transfer functions.

g. Continuous versus discrete models.

In a continuous model, a variable has a value at any given instant of time. In a discrete model, a variable has only values at discrete instances in time, for example, every second.

Discrete time models describe the state of the system at given time intervals and are therefore useful for efficient computation. When using computers for calculation of the model output, continuous models need to be discretized, since numerical solution methods require discrete models. Discretization of model equations refers to the approximation of (usually) the first and higher order derivatives in the model. There are several difference approximations that can be made, a forward difference approximation, a backward and a central difference approximation.

h. Time-invariant versus time-varying models

A time-invariant model does not change its parameters over time. For time-varying models, their parameters depend on time. It can be the functions of time, or perhaps some random variables in certain ranges.

## 2.2 Mass Relationship of Fluids

The model is used for a variety of applications, such as study of the dynamic behavior, process design, model-based control, optimization, controllability study, operator training, and prediction. Some modeling steps might require the knowledge of physical fundamentals, such as conservation balances. In this section the conservation balances for mass and momentum of fluids, in this case it is narrowed down to liquid and gas, are introduced.

### 2.2.1 Mass balance

For a liquid and gas buffer as shown in Figure 2.1 and Figure 2.2, the mass balance holds [38].

The following equation relates the rate of change in mass,  $m$ , to the difference between inlet mass flow,  $F_{m,in}$ , and outlet mass flow,  $F_{m,out}$ :

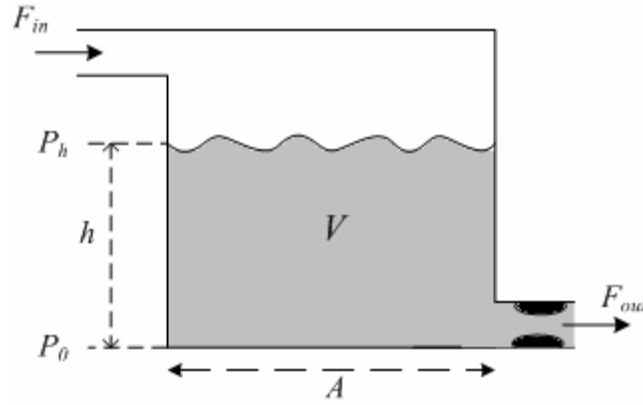
$$\frac{dm}{dt} = F_{m,in} - F_{m,out} \quad (2.1)$$

In case there are  $N$  inlet and  $M$  outlet flows, this equation may be written as:

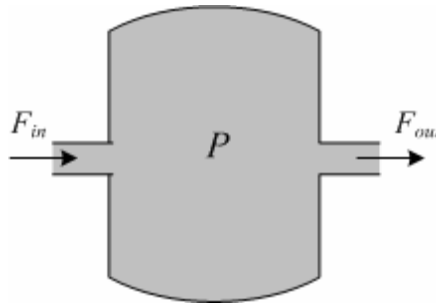
$$\frac{dm}{dt} = \sum_{i=1}^N F_{m,in,i} - \sum_{j=1}^M F_{m,out,j} \quad (2.2)$$

Often the volumetric flow is used instead of the mass flow, which is:

$$F_{m,i} = \rho_i F_{v,i} \quad (2.3)$$



**Figure 2.1:** Mass balance in a liquid tank with variable volume and fixed pressure.



**Figure 2.2:** Mass balance in a pressure chamber with variable pressure and fixed volume.

The relation in (2.3) is only useful if the density,  $\rho_i$ , and volumetric flow,  $F_{v,i}$ , are measured variables and if the densities of the flows are similar. Usually, the outlet flows

all have the same composition, similar to the composition inside the buffer. The mass balance in (2.2) can then be written as:

$$\frac{dm}{dt} = \sum_{i=1}^N \rho_i F_{v,in,i} - \sum_{j=1}^M \rho_{out,j} F_{v,out,j} . \quad (2.4)$$

The density  $\rho$  in (2.4) is defined as the mass per unit volume at certain pressure, temperature, and composition:

$$\rho = \left( \frac{m}{V} \right)_{P_0, T_0, x_0} . \quad (2.5)$$

This relationship holds at every instant in time. Therefore the rate of change in the mass of a system can be described by:

$$\frac{dm}{dt} = \rho \frac{dV}{dt} + V \frac{d\rho}{dt} . \quad (2.6)$$

Substituting (2.6) to (2.2) results in:

$$\rho \frac{dV}{dt} + V \frac{d\rho}{dt} = \sum_{i=1}^N F_{m,in,i} - \sum_{j=1}^M F_{m,out,j} . \quad (2.7)$$

The density depends on the state variables: the pressure  $P$ , temperature  $T$ , and composition  $x$ . In the operating point  $P_0, T_0, x_0$  it holds that:

$$\rho_0 = f(P_0, T_0, x_0) . \quad (2.8)$$

For changes in the state at operating point, the specific mass will change. For example, if the pressure rises, the density will increase. If the change in independent variables is small, in this case the pressure, the change in the dependent variable will depend linearly on the change in independent variable.

If the pressure or temperature changes, the change in relative density for a pure component can be calculated from the following equation:

$$\frac{\rho - \rho_0}{\rho_0} = \frac{1}{\rho_0} \left( \frac{\partial \rho}{\partial P} \right)_{P_0, T_0, x_0} (P - P_0) + \frac{1}{\rho_0} \left( \frac{\partial \rho}{\partial T} \right)_{P_0, T_0, x_0} (T - T_0) . \quad (2.9)$$

The partial derivatives are constant in the operating point:

$$\frac{\rho - \rho_0}{\rho_0} = \frac{1}{\beta}(P - P_0) - \gamma(T - T_0). \quad (2.10)$$

The constants in (2.10) are  $\beta$  and  $\gamma$ , they are valid for the operating point  $P_0, T_0, x_0$ .  $\beta$  is the isothermal bulk modulus (the inverse of the compressibility):

$$\beta = \rho_0 \left( \frac{\partial P}{\partial \rho} \right)_{P_0, T_0, x_0}, \text{ or } \beta = -V_0 \left( \frac{\partial P}{\partial V} \right)_{P_0, T_0, x_0}, \quad (2.11)$$

and  $\gamma$  is the thermal expansion coefficient, which describes the relationship in the change of volume with the change of temperature:

$$\gamma = \frac{1}{V_0} \left( \frac{\partial V}{\partial T} \right)_{P_0, T_0, x_0}, \text{ or } \gamma = -\frac{1}{\rho_0} \left( \frac{\partial \rho}{\partial T} \right)_{P_0, T_0, x_0}. \quad (2.12)$$

Hence for changes in the state variables  $P$  and  $T$  at constant composition, the changes in density can be described by:

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial P} \frac{dP}{dt} + \frac{\partial \rho}{\partial T} \frac{dT}{dt} = \rho \left( \frac{1}{\beta} \frac{dP}{dt} - \gamma \frac{dT}{dt} \right). \quad (2.13)$$

### 2.2.1.1 Liquid accumulation

For the accumulation of liquid in a vessel, as shown in Figure 2.1, the mass balance is described in (2.2). If there is only one inlet flow and one outlet flow, this equation can be written as:

$$\rho \frac{dV}{dt} + V \frac{d\rho}{dt} = F_{m,in} - F_{m,out}, \quad (2.14)$$

with the density  $\rho$  is a function of pressure and temperature, as described in (2.9).

The parameter  $\beta$ , which is defined in (2.11), can be used if the pressure change is gradual, such that the heat produced during this change can also be removed, ensuring that the temperature remains constant. Usually this is hard to realize, and a correction has to be applied. If the pressure change is fast, such that all heat remains in the system, the adiabatic bulk modulus should be used:

$$\beta_{ab} = \frac{c_p}{c_v} \beta, \quad (2.15)$$

in which  $c_p$  is the specific heat at constant pressure and  $c_v$  is the specific heat at constant volume. This modulus also accounts for the temperature effect. For liquids, the ratio  $c_p / c_v$  is slightly larger than 1. The expansion coefficient  $\gamma$  for most liquids is of the order of  $10^{-3} \text{ K}^{-1}$ . Exceptions are water vapor, ethylene oxide, and mercury.

If the temperature and pressure effects can be neglected, (2.14) can be reduced to:

$$\rho \frac{dV}{dt} = F_{m,in} - F_{m,out} . \quad (2.16)$$

If the inlet density is same with the outlet density, (2.16) becomes:

$$\frac{dV}{dt} = F_{v,in} - F_{v,out} . \quad (2.17)$$

### 2.2.1.2 Gas accumulation

For an ideal gas, it holds that [38]:

$$\rho = \frac{nM}{V} = \frac{PM}{RT} , \quad (2.18)$$

in which:

- $n$  number of moles
- $M$  molecular weight, kg/mole
- $V$  volume,  $\text{m}^3$
- $P$  absolute pressure,  $\text{N/m}^2$
- $R$  gas constant,  $\text{N.m/mole.K}$
- $T$  absolute temperature,  $\text{K}$ .

For an accumulation gas tank with fixed volume, as shown in Figure 2.2, with one inlet flow and one outlet flow, the mass balance is described in (2.14). As the gas buffer has a fixed volume,  $dV / dt = 0$ , hence:

$$V \frac{d\rho}{dt} = F_{m,in} - F_{m,out} . \quad (2.19)$$

From (2.18), the ideal gas law at constant temperature follows that:

$$\frac{d\rho}{dt} = \frac{M}{RT} \frac{dP}{dt} . \quad (2.20)$$

Substitution of (2.20) into (2.19) results in:

$$\frac{MV}{RT} \frac{dP}{dt} = F_{m,in} - F_{m,out} . \quad (2.21)$$

In (2.21), the ratio  $MV / RT$  represents the mass capacity,  $Cap_{m,gas}$ , of the gas at constant temperature in a fixed volume, hence:

$$Cap_{m,gas} \frac{dP}{dt} = F_{m,in} - F_{m,out} . \quad (2.22)$$

A similar equation can also be derived for the volumetric flow. At varying pressure but constant temperature and composition, it follows (2.13) that:

$$\frac{d\rho}{dt} = \frac{d\rho}{dP} \frac{dP}{dt} = \frac{\rho}{\beta} \frac{dP}{dt} . \quad (2.23)$$

Substitution (2.23) into (2.19) and (2.3) results in:

$$\frac{\rho V}{\beta} \frac{dP}{dt} = \rho_{in} F_{v,in} - \rho F_{v,out} . \quad (2.24)$$

If the densities are the same, (2.24) becomes:

$$\frac{V}{\beta} \frac{dP}{dt} = F_{v,in} - F_{v,out} . \quad (2.25)$$

In (2.25),  $V / \beta$  represents the compression capacity of the gas,  $Cap_{v,gas}$ , at a certain temperature and fixed volume. This equation is only valid in a certain operating range.

### 2.2.2 Fluid momentum balance

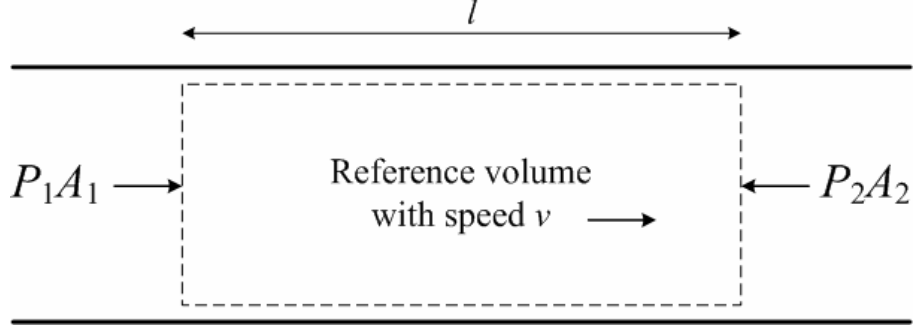
Flow induction is often called inertia, since this phenomenon is a result of liquid inertia.

For the reference volume in Figure 2.3, the momentum balance holds:

$$\sum F_{external} = \frac{dmv}{dt} = m \frac{dv}{dt} + v \frac{dm}{dt} , \quad (2.26)$$

with the external forces are:

$$\sum F_{external} = P_1 A_1 - P_2 A_2 . \quad (2.27)$$



**Figure 2.3:** Reference volume in a fluid pipe [38].

In case the pipe cross-sectional areas  $A = A_1 = A_2$ , (2.27) can be simplified to:

$$\sum F_{external} = A(P_1 - P_2) = A\Delta P. \quad (2.28)$$

Since the mass of the reference volume is constant, combination of (2.26) and (2.28) results in:

$$A\Delta P = m \frac{dv}{dt}. \quad (2.29)$$

Substitution of the equations for the mass  $m = \rho l A$  and the velocity  $v = F_v / A$  results in:

$$m \frac{dv}{dt} = \rho l A \frac{dF_v / A}{dt}. \quad (2.30)$$

Thus for a pipe with length  $l$ , it follows:

$$\Delta P = \frac{\rho l}{A} \frac{dF_v}{dt}. \quad (2.31)$$

### 2.2.3 Flow-through valves

The flow-through valves are often described by the following relationship [38]:

$$F_v = C_v f(u) \sqrt{\frac{\Delta P_v}{s.g.}}. \quad (2.32)$$

where:

- $F_v$  = volumetric flow,
- $C_v$  = valve coefficient,
- $u$  = fraction of valve opening,
- $\Delta P_v$  = pressure drop across the valve,

s.g. = specific gravity of the fluid,

$f(u)$  = fraction of the total flow area of the valve (varies from 0 to 1, as a function of  $u$ ).

Three common valve characteristics are: linear, equal-percentage, and quick-opening. For a linear valve, the fraction of the total flow area of the valve ( $f(u)$ ) as the function of the fraction of valve opening ( $u$ ) is given as

$$f(u) = u, \quad (2.33)$$

while for an equal-percentage valve

$$f(u) = \alpha^{u-1}, \quad (2.34)$$

with  $\alpha$  is constant, and for a quick-opening valve

$$f(u) = \sqrt{u}. \quad (2.35)$$

## 2.3 Dynamic Behavior of Series Processes

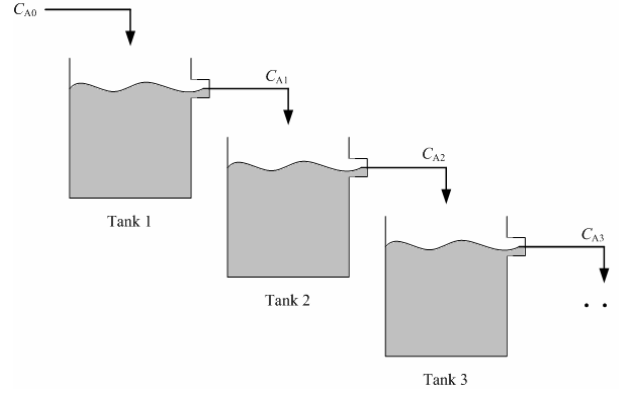
The overall dynamics of process plants with some interconnections or integrations can be very different from the dynamics of the individual units. For instance, material recycle and heat integration may dramatically alter the time constants of the plant, and may give rise to instability or oscillatory behavior (limit cycles), even when the individual units are stable by themselves. Moreover, plant interconnections may introduce fundamental limitations in the achievable performance of any control system [39].

One of the interconnection systems that commonly found in the plant is class of series processes. This section presents the dynamic behavior of series processes, which can be divided into two major categories: noninteracting series and interacting series.

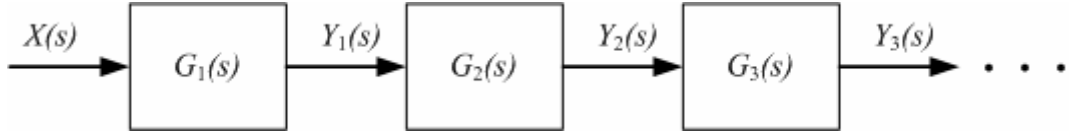
### 2.3.1 Noninteracting series process

An example of a noninteracting series process is chemical reaction process performed in several steps (tanks), as shown in Figure 2.4. The block diagram in Laplace domain of such a process is shown in Figure 2.5, where  $X$  represents the inlet concentration ( $C_0$ ),  $Y_1$  represents the outlet concentration of the first tank ( $C_1$ ),  $Y_2$  represents the outlet concentration of the second tank ( $C_2$ ), and so on.  $G_i(s)$  represents the individual transfer function of the  $i^{\text{th}}$ -tank.





**Figure 2.4:** A pH neutralization process with noninteracting series structure.



**Figure 2.5:** The transfer function of noninteracting series process.

Marlin [40] and Faanes & Skogestad [41] point out the important features of the system with noninteracting series structure as follows:

- The properties in one unit influence the properties in downstream units, but not the other way round. For example, in the process shown in Figure 2.4, the concentration in tank 2 does not affect the concentration in tank 1 but does affect tank 3.
- The model for the general noninteracting series of first-order systems can be developed by taking the Laplace transform of each unit and combining them into one input-output expression. For a series of systems shown in Figure 2.4, each represented by a transfer function  $G_i(s)$  in Figure 2.5, so the overall transfer function is:

$$\frac{Y_n(s)}{X(s)} = G_n(s)G_{n-1}(s)\cdots G_1(s) = \prod_{i=0}^{n-1} G_{n-i}(s). \quad (2.36)$$

For  $n$  first-order systems in series, this gives:

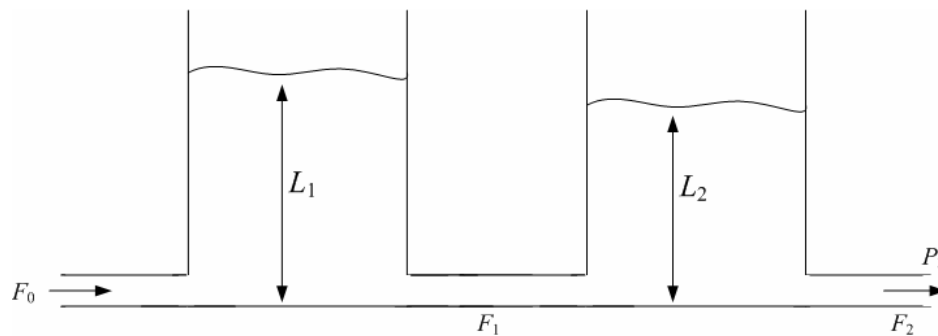
$$\frac{Y_n(s)}{X(s)} = \frac{\prod_{i=0}^{n-1} K_{n-i}}{\prod_{i=0}^{n-1} (\tau_{n-i}s + 1)}, \quad (2.37)$$

with  $K_{n-i}$  and  $\tau_{n-i}$  are the gain and the time constant for individual units respectively. Thus, the model of noninteracting systems can be determined directly from the individual models.

- c. If each system is stable (i.e.,  $\tau_i > 0$  for all  $i$ ), the series system is stable. This follows from the important observation that the poles (roots of the characteristic polynomial) of the noninteracting series system are the poles of the individual systems.

### 2.3.2 Interacting series process

The second major category of series processes is interacting processes. The example of an interacting process is the level-flow process shown in Figure 2.6.



**Figure 2.6:** A process with interacting series structure.

Marlin points out some important features of the interacting series processes as follows:

- a. The interactions between upstream and downstream happen in both ways round. Therefore, the downstream properties also affect the upstream properties, for example, for the process shown in Figure 2.6, the exhaust pressure ( $P_3$ ) influences both levels  $L_1$  and  $L_2$ .
- b. The model for the general interacting series process of first-order systems can be developed by deriving the transfer functions of each individual units and combining them into one input-output expression. However, it results in poles of the interacting systems that are different from the poles of the individual systems. So that, even if each individual units is stable, there is no guarantee that the overall process is also stable.

Since series structures are commonly found in process plants, there is a need to construct such systems in a suitable model that can be used in various applications, such as for model-based control. The procedure for deriving the model of noninteracting series processes that can be used for model-based control applications can be found in [41], however, the procedures for interacting series processes, has not been reported.

## 2.4 Multivariable Control System

The aim of control system is to force a given set of process variables to behave in some desired and prescribed way by either fulfilling some requirements of the time or frequency domain or achieving the best performances as expressed by an optimization index. The process engineer design the process according to the best of their knowledge in the field by assuming some operating conditions. Later on, the process will run under some other conditions that support external disturbances usually not well known or determined. Also the characteristic of the process will change with time and/or the load. It is the role of the control system to cope with these changes, also providing a suitable behavior. The scope of control task varies widely. The main goal is to keep the process running around the nominal conditions. In other cases, the control purpose will be to transfer the plant from one operating point to another or to track a given reference signal. In some other cases, the interest lies in obtaining the best features of the plant achieving, for instance, the maximum production, minimum energy consumption or pollution, or minimum time in performing a given task [42].

The process to be controlled is an entity of which the complexity can vary from simple to a very complex system. Independently of its design, carried out taking into account control requirements or not, control design assumes that the equipment modules are given and are already interconnected according to the guidelines of process experts. Sometimes, analysis of expected performance with a particular control system may advise changes in the process or instrumentations (sensors and actuators).

In order to control the process, some manipulated variables should be available, allowing introduction of control actions in the process to force it towards evolving in the desired way. In the kind of multivariable process, more than one manipulated variable is always available, providing more richness and options in controlling the process. These manipulated variables will act on the process through the corresponding actuators. To get

information about the process, some internal variables should be measured, being considered as output variables. Again, in the multivariable process, more than one output variable will be considered. The controlled variable could be these variables themselves or some other directly related to them.

The input variables or signals acting on the process but not being manipulated to achieve the control goals should be considered as disturbances. These disturbance can be predictable (deterministic) or not. Also, the disturbance can be measurable or not.

To study the behavior of a process is to analyze the various variables and their relationship. These variables can be [42]:

- a. External or inputs, being determined by other processes or the environment, acting on the process, and considered as:
  - i. Manipulated variables, if they are used to influence the dynamics of the process. Actuators will amplify the control commands to suitable power levels to modify plant's behaviors.
  - ii. Disturbances, if they are uncontrollable outputs of other subsystems, or non-manipulated inputs.
- b. Internal, being dependent on the process inputs, system structure, and parameters. They can be classified as:
  - i. Outputs or measured variables, if they are sensed and provide information about the process evolution.
  - ii. Controlled variables, if the control goals are based on them. They can be outputs or not, depending on the sensor's availability and placement.
  - iii. State variables, which are can be simply said as a minimum set of internal variables allowing the computation of any other internal variable if the inputs are known.

If the process is physically available and some experiments can be carried out, the dynamic behavior of the process can be captured and (partially) represented by a model using identification and parameter estimation techniques. However, it is not easy (or may be not plausible) to make a model that really represents the whole dynamics of the

system. To simplify the study, it is necessary to make some basic assumptions. Several priori assumptions are necessary to be made, especially in this work, such as: linearities, lumped parameters, and time invariance.

Multivariable control occurs in nearly all processes, because production rate, inventory, process environment, and product quality are normally controlled simultaneously. Some modifications to the PID feedback algorithm have been done to accommodate the requirements of multivariable control system. The wide applications of PID in feedback, cascade, and combined feedforward/feedback have indicated that the adoption of PID as the standard algorithm in 1940s was an appropriate choice. Perhaps the most remarkable feature of the PID is the success of this single algorithm in so many different applications. However, the development of the PID lacked a fundamental structure from which the algorithm could be derived, limitations could be identified, and enhancements could be developed.

The developments of advanced multivariable control, such as MPC, give a great insight into the roles of both the control algorithm and the process in the behavior of feedback systems. These developments also provide a method for tailoring the feedback control algorithm to each specific application. Because a model of the process is an integral part of the control algorithm, the controller equation structure depends on the process model, in contrast to the PID controller, which has only one equation structure [40].

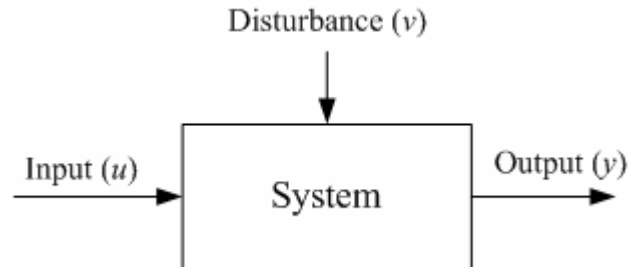
## 2.5 System Identification

As mentioned in section 2.1, the system can be modeled based on physical and chemical laws of conservation, or based on input-output data and only describe the relationship between the measured input and output data of the process. System identification provides the techniques to construct an empirical model from input and output data. This section presents the introduction of system identification with special attention to the class of state-space models.

### 2.5.1 Overview

The schematic diagram of a dynamic system with input  $u$ , output  $y$ , and disturbance  $v$  is shown Figure 2.7. The input  $u$  and the output  $y$  can be observed, but not the disturbance  $v$ . The input  $u$  can be directly manipulated, but not the output  $y$ . Even if the inside structure

of the system is unknown, the measured input and output data provide useful information about the system behavior. Thus, it can be used to construct a mathematical model that describes dynamics of the system [43].



**Figure 2.7:** A system with input, output, and disturbance.

System identification is a methodology developed mainly in the area of automatic control, by which the best model(s) can be chosen from a given model set based on the observed input-output data from the system. Hence, the problem of system identification is specified by three elements [43, 44]:

- a. A data set  $\mathcal{D}$  obtained by input-output measurements.

The input-output data  $\mathcal{D}$  are collected through experiment. In this case, the experiment has to be well designed, which requires the ability to determine the input or test signals, output signals to be measured, the sampling interval, etc., thereby systems characteristics are well reflected in the observed data. Thus, to obtain useful data for system identification, it is necessary to have some priori information or physical knowledge about the system.

- b. A model set  $\mathcal{M}$ , or a model structure, containing candidate models.

A choice of model set  $\mathcal{M}$  is a difficult issue in system identification, but usually several classes of discrete linear time-invariant (LTI) systems is used. Since these models do not necessarily reflect the knowledge about the structure of the system, they are referred to as black box models. One of the most difficult problems is to find a good model structure, or to fix orders of the models, based on the given input-output data.

- c. A criterion  $V_N$  to select the best model(s), or a rule to evaluate candidate models, based on the  $N$  number of data.

To find a model in the model set  $\mathcal{M}$ , a criterion to measure the distance between a model and a real system is needed. In terms of the input  $u$ , the output of a real system  $y$ , and the model output  $\hat{y}$ , the criterion is usually defined as:

$$V_N = \sum_{k=0}^{N-1} l(y(k), \hat{y}(k), u(k)), \quad (2.38)$$

where  $l(\cdot)$  is a nonnegative loss function, and  $N$  the number of data. If the model set is parametrized as  $\mathcal{M} = \{\mathcal{M}_v, v \in \Upsilon\}$ ,  $\Upsilon$  is a set of model parameters, then the identification in narrow sense reduces to an optimization problem minimizing the criterion  $V_N$  with respect to  $\Upsilon$ .

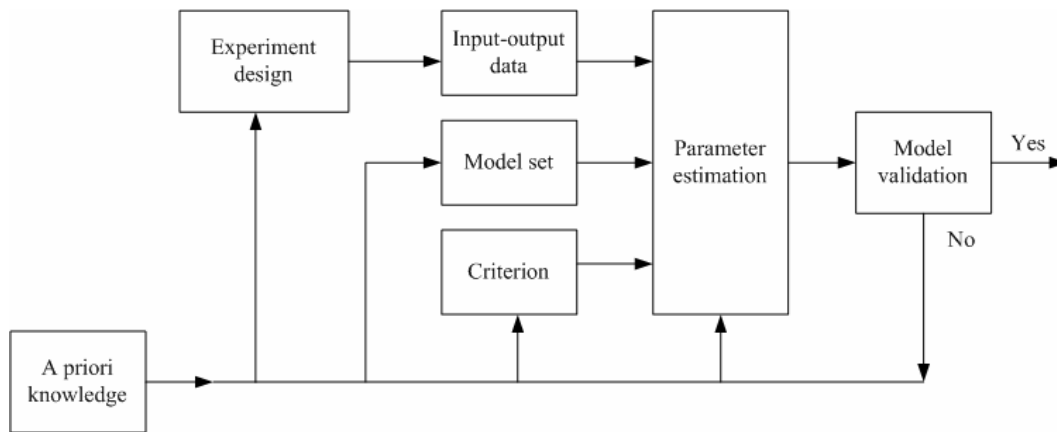
Given three basic elements in system identification, the best model  $\mathcal{M}^* \in \mathcal{M}$  can be found theoretically if the following are satisfied:

- a. A condition for the existence of a model that minimizes the criterion.
- b. An algorithm of computing models.
- c. A method of model validation.

Model validation is used to determine whether or not an identified model should be accepted as a suitable description that explains the dynamics of a system. Thus, model validation is based on the way in which the model is used, such as the fitness of the model to real data.

A flow diagram of system identification is shown in Figure 2.8. It can be seen that the system identification procedure has an iterative or feedback structure.

Models obtained by system identification are valid under some prescribed conditions. For instance, the models are valid for a certain neighborhood of working point. The models obtained by system identification also do not provide a physical insight into the system because parameters in the model have no physical meaning. As shown in Figure 2.8, a desired model cannot be obtained unless the identified models are iteratively evaluated on several model structures, model orders, etc. [43].



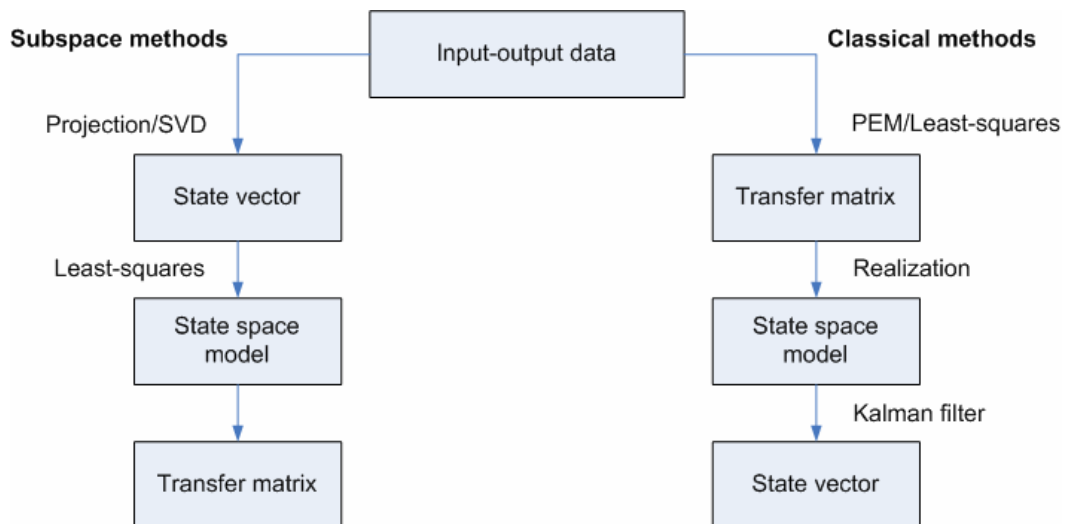
**Figure 2.8:** A flow diagram of system identification [43, 44].

### 2.5.2 System identification for state-space models

There are two main approaches for constructing state-space model, as it is shown in Figure 2.9, where the left-hand side is the subspace method, and the right-hand side is the classical optimization-based method. In the classical methods, such as prediction error method (PEM), a transfer function model is first identified, and then a state-space model is obtained using some realization technique. From the state-space model, the state vectors or the Kalman filter state vectors then can be computed. In subspace methods, however, the state estimates are first constructed from given input-output data by using simple procedure based on tools of numerical linear algebra, and a state-space model is obtained by solving a least-squares problem. If necessary, the transfer matrix then can be computed easily.

Overschee and De Moor [45] point out that an important achievement of the research in subspace identification is to demonstrate how the Kalman filter states can be obtained from input-output data using linear algebra tools, i.e., QR decomposition and singular value decomposition. An important consequence is that, once these states are known, the identification problem becomes a linear least square problem in the unknown system matrices. This implies that one possible interpretation of subspace identification algorithms is that they conditionally linearized the problem, which, when written in the form of PEM, is a highly nonlinear optimization problem. Yet another point of view is that subspace identification algorithms do not identify input-output models, but they identify input-state-output models.





**Figure 2.9:** Subspace and classical methods of state-space system identification [43, 45].

### 2.5.3 Advantages and challenges

Fassois and Rivera [46] point out that practical interest in system identification is driven by difficulties associated with deriving models from physical principles. Even when a model structure is available from physical principles, it is often necessary to apply the system identification techniques to estimate parameter values. Unlike physical modeling, the system identification develops the empirical models of systems based on excitation and response measurements, and therefore, its advantages can be simply addressed to its capability to overcome several obstacles in the physical modeling. Bohlin [47] notes that beside the time consuming, some well-known obstacles of the physical modeling in practice can be outlined as follows:

- a. Unknown relations between some variables.

Engineers often do not have the complete mathematical knowledge of the object to be able to construct the model.

- b. Too many relations for convenience.

When there is knowledge about the system being modeled, the result is often too complex to be possible to simulate with the ease required for parameter fitting. Many physical phenomena are describable only by partial differential equations. Simulation would then require high computational capability.

c. Unknown complexity.

It depends solely on the designer to determine how much of the relations to include in the model.

d. Model validation.

It is also difficult to validate the physical model with the measurements data, especially when the model parameters are determined based on the literature or from separate experiments.

From engineering viewpoint, the complexities and difficulties in the physical modeling mentioned above are often too involved to be really useful [45]. For engineers, the exact model of a system is not really of interest. The attention is commonly more in the potential engineering applications of models. The system identification techniques offer the methods that eliminate such difficulties. The quality of a model is dictated by the ultimate goal it serves. Model uncertainty is allowed as long as the robustness of overall system is ensured. Overschee and De Moor [45] point out that the system identification techniques trade-off model complexity versus accuracy.

System identification is characterized by a strongly interdisciplinary nature that draws from systems theory, signal processing, optimization, and statistics [46]. The system being modeled and the applications of models also vary widely. Therefore, the challenges associated with applying system identification in practice cannot be limited in a single monolithic task. Focused on its applications for monitoring and control purposes of process plants, the challenges in system identification can be clustered into several major problems. Jørgensen and Lee [18] present the recent challenges in the applications of system identification techniques in process plants, which are addressed as the gap between research and practice, as follows:

a. Plant testing.

Commonly, industrial plant testing use simple input signals, like step or PRBS signals. In addition, it is almost always limited to perturbing one input at a time, mostly out of the concern for unpredictable effects on process behavior. Many literatures present the optimal test signal design methods including those that attempt to incorporate specific control requirements and process characteristics into design in

an iterative manner. However, such tailored and iterative designs have rarely been attempted in practice, if ever.

b. Model structure.

In practice, the popular structures are finite impulse response (FIR) models and Auto Regressive with eXogenous (ARX) models, both of which lead to linear regression problems. Other structures, like Auto Regressive Moving Average with eXogenous (ARMAX) models, Output Error (OE) models, and Box-Jenkins models, which require non-convex optimization, are less common but are used in some occasions.

In some cases, multiple-input single-output (MISO) structures are used, in which a separately parameterized model is fitted for each output. According to Jørgensen and Lee [18], this practice is inefficient compare to the use of multiple-input multiple-output (MIMO) structures, in view that the fact most industrial process outputs exhibit significant levels of cross-correlation. However, several researches [19, 21] also note that, for certain cases, MISO identification remains relevant. Hence, the preference of model structure remains in deliberation.

c. Model estimation.

So far, the most dominant method for estimating model parameters is by using PEM. Perhaps, this is owing to its flexibility and sound theoretical basis as well as the ready availability of software tools. However, with multivariable structures, PEM requires special parameterizations and non-convex optimization, a fact that perhaps explain the industry's proclivity toward use of MISO structures. The subspace approach is designed to obviate these problems but requires relatively large data sets.

The statistical methods such as the maximum likelihood estimation or Bayesian estimation have not found much use. This is probably because of the lack of probability information or the fact that these complex methods often reduce to the same least squares calculation as PEM under commonly made probability assumptions. Similarly, the use of nonparametric methods such as the frequency-domain empirical transfer function estimation seems rare, probably due to the lack of sufficiently large data sets.

d. The use of estimated model.

The PEM approaches with many structures such as ARX or ARMAX, as well as the subspace approach, yield a combined deterministic-stochastic system model. However, the noise part of the model is seldom used in control system design. This may be due to the belief that disturbances experienced during an identification experiment are not representative of those encountered during real operation. However, this practice bears some danger as the two model parts are identified to work together as a single predictor. Also, many process monitoring and soft-sensing schemes require precisely information on how variables are correlated to one another in time due to unknown inputs and noises. Hence, the noise part of a multivariable model, when fitted with appropriate data, can serve a very useful function. In addition, some control applications, such as those involving inferential estimation, should clearly benefit from availability of an accurate noise model.

e. Model validation.

The standard tests such as residual analysis and cross validation are widely used in the industrial practice. However, model uncertainties are seldom captured in a form that can be used for robust controller design.

f. Closed-loop testing and iterative improvement.

Many industrial processes have already in place one or more loops that cannot be removed for safety and/or economic reasons. Hence, it is suspected that closed-loop testing has already been practiced to some extent. However, it is not clear whether practitioners are always aware of potential problems that can arise with usage of feedback correlated data. The idea of continually improving the closed-loop performance by using data collected from a working loop is also very attractive from a practical viewpoint. It may connect well with the industry's growing concern over maintaining performance of advanced controllers.

g. Nonlinear identification.

Several systematic methods for identifying first engineering principle models are available. Some implementations of nonlinear model predictive control have also been reported. The most common industrial approach to deal with process nonlinearities is by use of multiple models. Switching rules among different models

are *ad hoc* and seldom systematically designed. Some applications of artificial neural networks are reported but their effectiveness as *causal* models, e.g., as optimization and control would use them, is questionable at best. Despite the vigorous research in this area during past decade, the field still lacks a basic framework and a unifying theoretical foundation.

## 2.6 Model Predictive Control

Model predictive control refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of a plant. MPC was originally developed to meet the specialized control needs of petroleum refineries. Nowadays, it has been used in a wide variety of application areas including chemicals, food processing, automotive, aerospace, metallurgy, and power plant [48].

Model predictive control has been known to be one of the leading advanced control algorithms in industrial practice. One of its advantages includes the ability to explicitly handle constraints in inputs, states and outputs. An explicit internal model is used for the online optimal control calculation at each sampling time while considering the process behavior over a future, finite time horizon [49-52].

An MPC results in a linear control law which is easy to implement once the controller parameters are known. The derivation of the MPC parameters requires, however, some mathematical complexities. Although this is not a problem for people in the research control community where mathematical packages are normally available, it may be discouraging for those practitioners used to much simpler ways of implementing and tuning controllers [53].

Model predictive control has influenced process control practice significantly during the past 20 years. Surveys by Qin and Badgwell describe MPC products has been available since past 10 years [51]. These surveys summarize product features that relate to important issues in process control practice. Even the theoretical insight into MPC increased rapidly, but actual practice evolved more slowly [52-55].

### 2.6.1 Historical perspective

According to Froisy [54] and the survey by Qin and Badgwell [51], the history of industrial MPC can be outlined as follows:

The earliest efforts on industrial MPC could be dated back to Cutler and Ramaker [56] and Richalet et al. [57]. The initial work was probably being done at about the time of Chemical Process Control (CPC) 1 in 1976 [54]. A long sequence of MPC-oriented papers have appeared at every CPC since then, including one on Model Algorithmic Control (MAC) at CPC 2 in 1981 published by Mehra et al. [58]. Five years later, at CPC 3 in 1986, a session was devoted to MPC with focus on industrial applications. Although rather simple by current standards, the applications were clearly economically significant. Two papers from Froisy and Richalet [59] and Garcia and Prett [60] foretold of new directions for Identification and Command (IDCOM) and Dynamic Matrix Control (DMC), the notable early MPC variants [54]. An important theme was the need to address a wide range of real-world issues such as multiobjective and economic optimization, dealing with hard and soft constraints, and robustness to modeling errors.

There were two sessions on MPC at CPC 4 in 1991, attesting to its growing popularity. Some papers described industrial applications, while others from the academic community addressed theoretical issues such as stability of constrained MPC. The papers from Morari and Lee [61] and Ricker [62] dissected industrial MPC approaches, essentially reverse-engineering them in terms of traditional control engineering theory. The analysis comparing theory and practice, notably with practice as the first priority, was an important step. It exposes deficiencies both in practice and theory, thereby helping set new research directions. MPC was interpreted in a state-space framework, an important step needed to mold MPC with established theory such as linear quadratic control. However, MPC practice was largely formulated as a constrained optimization problem meaning that significant gaps still existed between theory and practice.

By the time of CPC 5 in 1996, the MPC session had a “recent advances” paper and one on nonlinear MPC. Also notable was a survey and overview of industrial MPC by Qin and Badgwell [63] describing various commercially available MPC products. These products which had largely evolved during the previous 10 years tended to be descendents of IDCOM and DMC. A few such as Shell Multivariable Optimizing Controller (SMOC) and Robust Multivariable Predictive Control Technology (RMPCT) were not direct descendants.

SMOC has the distinction of using state-space models and feedback via a Kalman filter, unlike the other cited products. More recent products, such as one from ABB, also use a

state-space formulation. The existence of several products confirmed the commercial significance and maturity of MPC technology. A detailed look at the survey exposes the fact that these products had ad hoc features needed to deal with inherent limitations, such as those noted by the academic community. Lack of an explicit disturbance model, special parameters for models with integrators and various parameters to deal with the finite model dynamic move plan calculation are examples.

A notable paper was published between CPC 4 and CPC 5 in 1993 by Muske and Rawlings [64]. It provides a framework for infinite horizon, constrained MPC that matched real-world needs fairly well. For example, a structure for disturbance models that meets important requirements and demonstrates nominal stability for constrained control was described. Another important CPC 5 paper by Wright [65] shows how to formulate a general (linear) MPC optimization problem for efficient computation.

There were no specific MPC sessions at CPC 6, but related papers appeared among other sessions. Eastman Chemical Company documented its experience with infinite horizon state-space MPC, presented by Vogel and Downs [66]. Their implementation was based on the Muske and Rawlings paper in 1993. Several papers addressed MPC with nonlinear models, including one that describes development of the nonlinear model predictive control (NLMPC) algorithm at ExxonMobil Chemical Company in conjunction with university and private collaborators by Young et al. [67].

### **2.6.2 Advantages and challenges**

There are extensive literatures presenting the advantages of MPC technology over other methods, amongst which that stands out are [49, 50, 52, 68, 69]. Based on these literatures, the advantages of MPC can be briefly summarized as follows:

- a. Because the concepts are very intuitive and at the same time the tuning are relatively easy, MPC is particularly attractive to staff with only a limited knowledge of control.
- b. MPC can be used to control a great variety of processes, from those with relatively simple dynamics to other more complex ones, including systems with long delay times or of non-minimum phase or unstable ones.
- c. The multivariable case can easily be dealt with MPC.

- d. MPC intrinsically has compensation for dead times.
- e. MPC introduces feedforward control in a natural way to compensate for measurable disturbances.
- f. MPC has the extension to the treatment of constraints which is conceptually simple and can be systematically included during the design process.
- g. MPC can take account of actuator limitations.
- h. MPC is very useful when future references (such as robotics or batch processes applications) are known.
- i. MPC is totally open methodology based on certain basic principles which allow for future extensions.

Despite its advantages and popularity, a number of issues in the MPC applications remain. Many recent researches have extensively explored the robustness of MPC to uncertainty in the process model and disturbance [13-17, 70], actuator limitations [49, 71], and computational complexity [72, 73], while other researches have focused on its real-time applications [23, 48, 52, 53, 55, 74].

Notably, there are quite substantial literatures on system identification approach to develop an empirical model of a system and quite well developed literatures on implementing a model predictive control algorithm in process plant. However, few explore the issues on interacting series problem adopted to sytem identification and model predictive control implementation. Considering the defining benefits of modeling, analysis, and controller design of this type of process, the motivation to embark on this work is based on the potential for extending and scaling the research in the interacting series process using the methodology to be presented in the subsequent chapters.



## Chapter 3

# System Identification of Interacting Series Process with Nonlinear Dynamics

This chapter presents the empirical modeling of the gaseous pilot plant which is a kind of interacting series process with presence of nonlinearities. In this study, the discrete-time identification approach based on subspace method with N4SID (Numerical algorithm for Subspace State-space System Identification) algorithm is applied to construct the state-space model around a given operating point. During plant testing, the system is probed in open-loop with variation of input signals. Three practical approaches are used and their performances are compared to obtain the most suitable approach for modeling of such a system. Performance comparison of three practical approaches are conducted using a fresh data set, which has not been used in the identification processes, to obtain the MIMO state-space model of such a system which is expected to be robust against the nonlinearities in the plant. Some important issues in the identification process and the construction of the linear MIMO state-space model of an interacting series process from input-output data using a linear system identification technique are also presented.

### 3.1 Introduction

In many instances, a dynamic system analysis and design requires a suitable mathematical model. Basically, there are two options to construct a mathematical model of a dynamic system. Scientist will be interested in models (physical laws) that carefully explain the underlying essential mechanisms of observed phenomena and that are not falsified by the available experiments. The necessary mathematical equipment is that of nonlinear partial differential equations. This is an analytical approach, which rigorously develops the model from first principles. For engineers however, especially in the industrial community, this framework is often much too involved to be really useful. The reason is that engineers are not really interested in the exact model as such, but more in the potential engineering applications of models. In this perspective, a mathematical model is only one step in the global design of a system. The quality of a model is dictated by the ultimate goal it serves. Model uncertainty is allowed as long as the robustness of the

overall system is ensured. Engineers, in contrast with scientist, are prepared to trade-off model complexity versus accuracy [45].

Nowadays, there are several mathematical modeling approaches being used in research communities as well as in practical applications. One of the most widely used is a class of state-space models. Based on the experiences, many industrial processes can be described very accurately by this type of models. Moreover, by now, the number of control system design tool that are available to build a controller based on this type of models, is almost without bound. For multiple-inputs multiple-outputs (MIMO) systems, the state-space representation seems as the only model that is conveniently to work with computer-aided control system design (CACSD). Most optimal controllers can be effectively computed in terms of the state-space model, while for other system representations, such as transfer function or polynomial, the calculations are not so elegant [45]. System identifications is a subclass of state-space modeling approach and basically it deals with the problem of obtaining “approximate” models of dynamic system from measured input-output data. This issue is of interest in a variety of applications, ranging from chemical process simulation and control to identification of vibration modes in flexible structures. Two of the methodologies in which dynamic systems have been modeled using the system identification are the prediction error method and subspace method. The prediction error method (PEM) is one of the most traditional system identification techniques. The PEM has excellent statistical properties provided the “true” PEM estimate can be found [75]. Subspace methods have their origin in state-space realization. Subspace identification method is a technique that has been developed since the late 80’s. It has attracted much attention, owing to its computational simplicity and effectiveness in identifying dynamic state-space linear multivariable systems. These algorithm are numerically robust and do not involve nonlinear optimization techniques [75]. However, there is no well-established rule tied to this approach for determining the structural parameter [76]. One of the most popular subspace identification methods is the N4SID developed by Overschee and Moor [45].

A discussion on the open-loop responses of the systems with series structures is given by Marlin [40] which specified them into two categories: noninteracting and interacting series. The main difference between these categories is about how the downstream properties influence the upstream properties. For noninteracting series systems, the states

in one process unit influence the states in the downstream unit, but not the other way round. One example of such a system is pH neutralization performed in several tanks in series [77]. Morud and Skogestad [39] note the poles and zeros of the transfer function of a noninteracting series process are the poles and zeros of the transfer functions of the individual units. Thus, the overall responses may be predicted directly from the individual units. This also implies that if each system is stable, the series system is stable.

In contrast with noninteracting systems, the downstream properties in interacting series processes influences upstream properties. Marlin [40] outlines the procedure for deriving overall transfer function of an interacting series system which is somewhat more complex than for a noninteracting system. Interestingly, the poles of the interacting system are different from the poles of the individual systems. The dynamic behavior of the interacting process has to be determined from the analysis of the overall transfer function, not the individual units.

The use of system identification to develop the empirical linear model of processes with series structures have been reported by several researches. In their paper, Gatzke et al. [78] perform the parametric identification process of a quadruple tank using subspace system identification method. Such a system has series structure with recycles. As the input signals, the pseudo-random binary sequence (PRBS) is used. The identification process is carried out without taking into account the prior knowledge of the process, and no assumption are made about the state relationships or number of process states. Only the number of the states used in the resulting process model was determined. The developed model is then used for model-based control using Internal Model Control (IMC). Modeling is also performed using step tests and Aspen software for use with Dynamic Matrix Control (DMC). Weyer [79] presents the empirical modeling of water level in an irrigation channel using system identification technique taking into account the prior physical information of the system. The model structure is derived from mass and momentum balances that employed nonlinear equations. The identified process is a kind of interacting series process, however, the model only has a single output variable which is the water level in the downstream of irrigation channel. The developed model has a multiple-inputs single-output (MISO) structure. Sotomayor et al. [75] present the multivariable identification of an activated sludge process benchmark, which can be categorized as a system that has series structure with recycle, using subspace-based

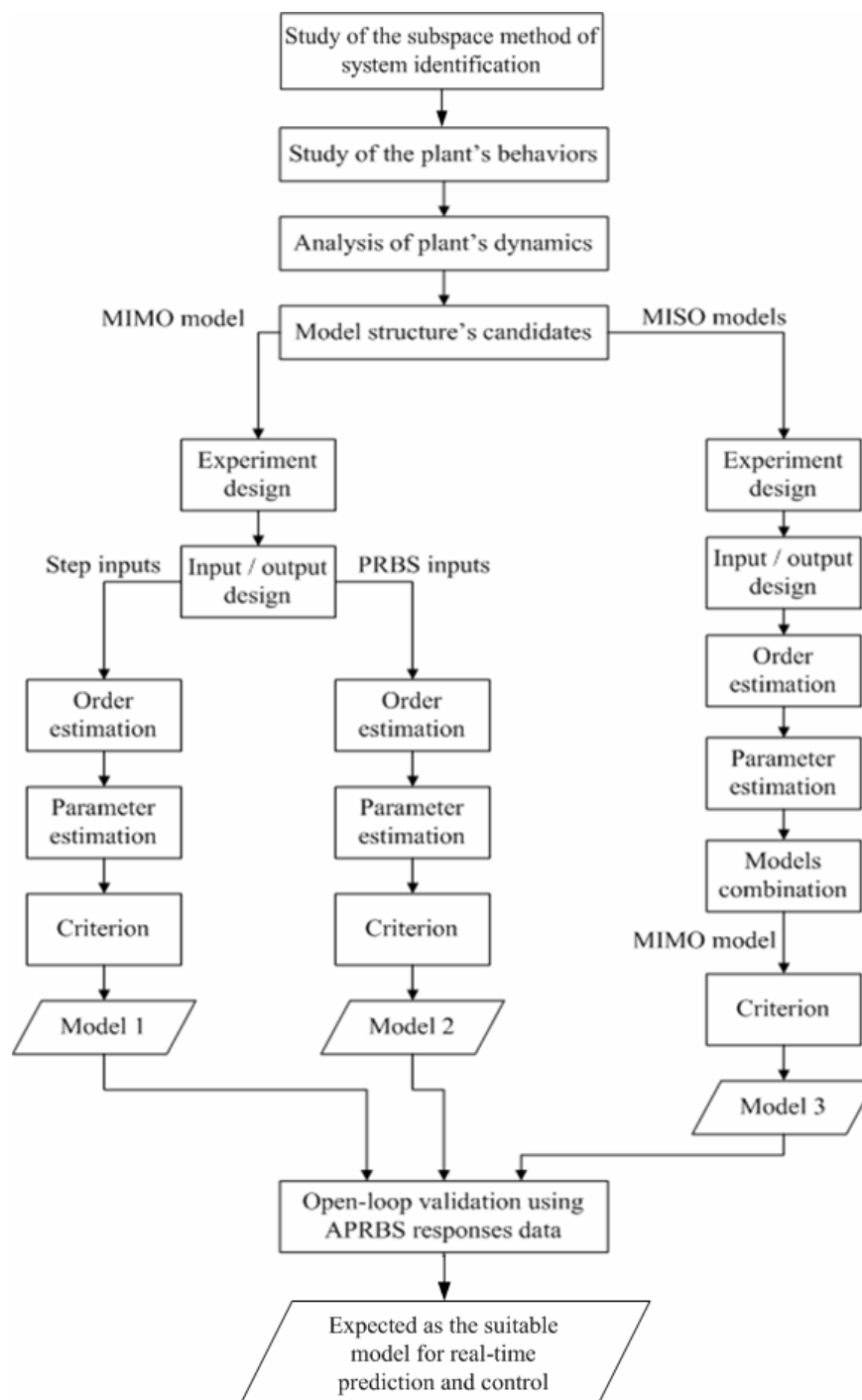
algorithms. Six subspace algorithms are used and their performances are compared to obtain the best model. A discrete-time identification approach based on subspace methods is applied to estimate a nominal MIMO state-space model. The interesting result is that the selected state-space model is a very low-order one and it described the complex dynamics of process well. However, a drawback of the methods used is that the physical insight of the process in the models is lost, a characteristic of black box model.

This work aims at identifying a LTI with lumped parameters state-space model of gaseous pilot plant that has a typical structure of an interacting series process where there exist strong influences between upstream and downstream variables. The process is also showing some nonlinearities either in the overall responses, such as the shiftiness of output variables, or in the responses of individual units from the respective inputs. The limited available measurements presence another challenge since the internal states of the system such as inlet, outlet, and internal flowrates are unknown. This work is an extension of the study on the practical approaches for constructing a MIMO state-space model from input-output data using a linear system identification technique that has appeared in [80]. In the paper [80], the authors have pointed out that the subspace identification method using N4SID algorithm was a more suitable method for gaseous pilot plant rather than PEM method, indicated by smaller identification and validation errors. In this work, the focus has been addressed to develop the proper procedure and method for constructing an empirical model of the interacting series processes from input-output data using system identification technique.

This chapter is organized as follows. Section 3.2 presents the methodology used in this work to develop the empirical model of an interacting series process using system identification technique. Section 3.3 introduces the subspace method of system identification. Section 3.4 presents the overview of an interacting series plant used in this study, i.e., the gaseous pilot plant. Some analyses of the plant dynamics and its possible model structures are also given in this section. Section 3.5 discusses the experimental designs, model structures and variable selections, order estimations, model validations, and the technique to merge several MISO models into a single MIMO state-space model. The analysis and evaluation of the developed models are given in section 3.6. Finally the summary is given in section 3.7.

### 3.2 The Methodology of Modeling and System Identification

The flow diagram of the methodology used in the model development that would be evolved in this investigation is given in Figure 3.1. For the purpose of the study, a lab-scale pilot plant that has the typical characteristics of the interacting series processes, i.e., the gaseous pilot plant, would be used.



**Figure 3.1:** The flow diagram of plant modeling and system identification.

As illustrated in Figure 3.1, this research would begin with the study of subspace method of system identification using N4SID algorithm that would be used to estimate the model parameters of the plant. The work would be continued with the examination of the plant behaviors and followed by the physical analysis of the plant dynamics. This includes the study of the plant responses from the external available inputs. From the analysis of plant dynamics, two kinds of model structure would be proposed. The first model structure would be treated with two different kinds of input signals, while another model structure would only be perturbed with one kind of input: step signal, resulted in three different state space models. The open-loop validations would then be performed using a fresh data set which has not been used in the identification procedures. The model which gives the best performance in open-loop validation would be expected as the most suitable model of such a plant, and more importantly, expected to be robust against the plant nonlinearities when it goes to the real-time implementation of the model-based control system.

Details of the model development and analysis of interacting series process described above are presented in these subsequent sections.

### 3.3 Subspace Method of System Identification

In discrete-time domain, a linear time-invariant system can be formed as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{\Lambda}\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{v}(k) \end{aligned} \quad (3.1)$$

where  $\mathbf{y}$  is the output vector,  $\mathbf{u}$  the input vector,  $\mathbf{x}$  the state vector,  $\mathbf{w}$  and  $\mathbf{v}$  are the innovation vectors, commonly refer to the plant noise and the measurement noise respectively, with zero mean and covariance matrix  $\mathbf{R} > 0$ , and  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$ , and  $\mathbf{\Lambda}$  are the coefficient matrices of appropriate dimensions. The unknown parameters in the state-space model are contained in these system matrices and covariance matrix  $\mathbf{R}$  of the innovation process.

Subspace identification methods are based on the following idea. Suppose that estimates of a sequence of state vectors of the state-space model of (3.1) are somehow constructed from the input-output data. Then for the prediction error  $\mathbf{\varepsilon}(k)$ ,  $k = 0, 1, \dots, N-1$ , its relation can written as

$$\begin{bmatrix} \hat{\mathbf{x}}(k+1) \\ \mathbf{y}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \mathbf{u}(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\eta}(k) \\ \mathbf{v}(k) \end{bmatrix} \quad (3.2)$$

where  $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$  is the estimate of state vector,  $\mathbf{u} \in \mathbb{R}^{n_u}$  the input vector,  $\mathbf{y} \in \mathbb{R}^{n_y}$  the output vector, while  $\boldsymbol{\eta}$  and  $\mathbf{v}$  are residuals. Since all the variables are given, (3.2) is a regression model for system parameters  $\boldsymbol{\Theta} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \in \mathbb{R}^{(n_x+n_y) \times (n_x+n_u)}$ . Thus the least-squares estimate of  $\boldsymbol{\Theta}$  is given by

$$\boldsymbol{\Theta} = \left( \sum_{k=0}^{N-1} \begin{bmatrix} \hat{\mathbf{x}}(k+1) \\ \mathbf{y}(k) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}^T(k) \\ \mathbf{u}^T(k) \end{bmatrix}^T \right) \left( \sum_{k=0}^{N-1} \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \mathbf{u}(k) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}^T(k) \\ \mathbf{u}^T(k) \end{bmatrix}^T \right)^{-1}. \quad (3.3)$$

This class of approaches are called the direct N4SID methods. This estimate uniquely exist if the rank condition

$$\text{rank} \begin{bmatrix} \hat{\mathbf{x}}(0) & \hat{\mathbf{x}}(1) & \cdots & \hat{\mathbf{x}}(N-1) \\ \mathbf{u}(0) & \mathbf{u}(1) & \cdots & \mathbf{u}(N-1) \end{bmatrix} = n_x + n_u \quad (3.4)$$

is satisfied. The covariance matrices of residuals are given by

$$\begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{R} \end{bmatrix} = \frac{1}{N} \sum_{k=0}^{N-1} \begin{bmatrix} \boldsymbol{\eta}(k) \\ \mathbf{v}(k) \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}^T(k) & \mathbf{v}^T(k) \end{bmatrix}. \quad (3.5)$$

Related to (3.1), it is assumed that the system is asymptotically stable, the pair  $(\mathbf{A}, \mathbf{C})$  is observable and the pair of  $(\mathbf{A}, \mathbf{B})$  is controllable [43, 45, 75].

The N4SID method is geometric in their approach, and involve the subspaces spanned by the rows and/or columns of matrices constructed from the input-output data. The general algorithm of the subspace methods involves three major steps [81, 82]:

- a. Unlike the more ‘classical’ estimation methods, such as prediction error method, based on a curve-fitting to derive polynomial models, the N4SID technique relies on subspace fitting strategies for the approximation of the extended observability matrix,  $\boldsymbol{\Gamma}_i$ , and/or the state sequence,  $\mathbf{X}_i$ , which are defined by

$$\boldsymbol{\Gamma}_i = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{i-1} \end{pmatrix}, \quad (3.6)$$

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{x}_i & \mathbf{x}_{i+1} & \mathbf{x}_{i+2} & \cdots & \mathbf{x}_{i+j-1} \end{pmatrix}. \quad (3.7)$$

- b. Secondly, a singular value decomposition of the previously estimated matrix is computed to estimate the order  $n$  of the state-space model.
- c. The final step is computing the matrices **A**, **B**, **C**, and **D** by solving over-determined sets of linear equations owing to least-squares or total least-squares computation techniques.

### 3.4 The Gaseous Pilot Plant as an Interacting Series Process

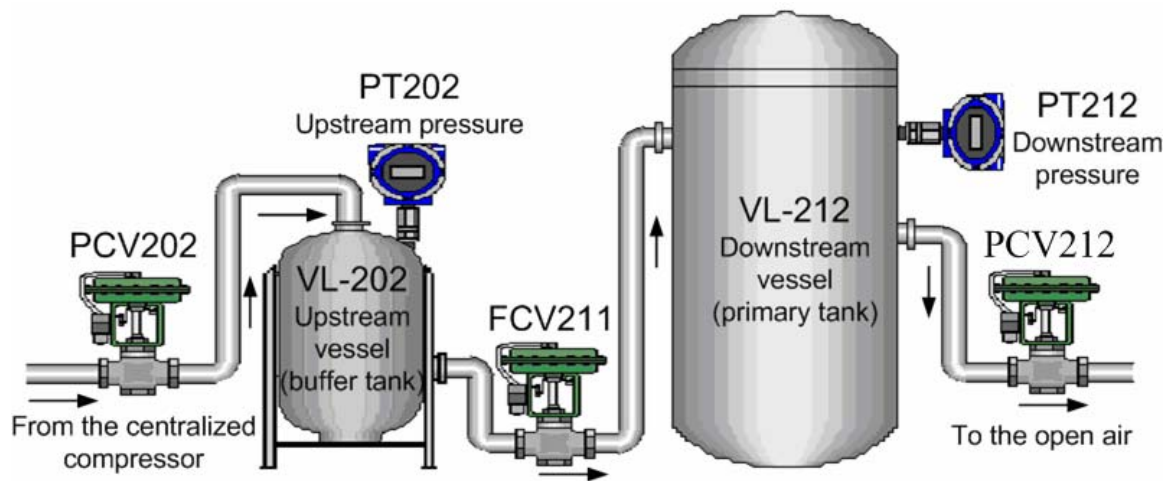
#### 3.4.1 Plant overview

To investigate the empirical modeling of an interacting series process, a lab-scale pilot plant that able to demonstrate the dynamics of gas inside the vessel and pipeline is used. This plant has a continuous air as feed, which is generated from a centralized compressor, and the pressure is maintained at 7 barg. There will be some pressure drop when the gas passes the inlet control valve (PCV202) and when enters the buffer tank (VL-202). The pressure will drop slightly further when the gas passes through the middle control valve (FCV211) and when it enters the primary tank (VL-212). These pressure balances are also affected by the opening of outlet control valve (PCV212). The pressure inside the buffer tank, which is also called the upstream pressure, is indicated by pressure transmitter PT202, while the pressure inside the primary tank, which is also called the downstream pressure, is indicated by pressure transmitter PT212. The schematic diagram of the gaseous pilot plant is shown in Figure 3.2.

The control objective is to maintain pressure balance inside the primary tank as well as the buffer tank by manipulating the control valves. This requires the construction of a suitable mathematical model that able to describe its dynamics. The state-space model is chosen, since the control system that will be used is a class of model-based control that much involves the state-space system and matrices manipulations.

During normal operating conditions, the operating range of pressure inside the buffer tank is from 4 to 6.5 barg while the pressure inside the primary tank will varies from 3.5 to 6 barg. Figure 3.3 shows a photograph of the gaseous pilot plant being used in this research.



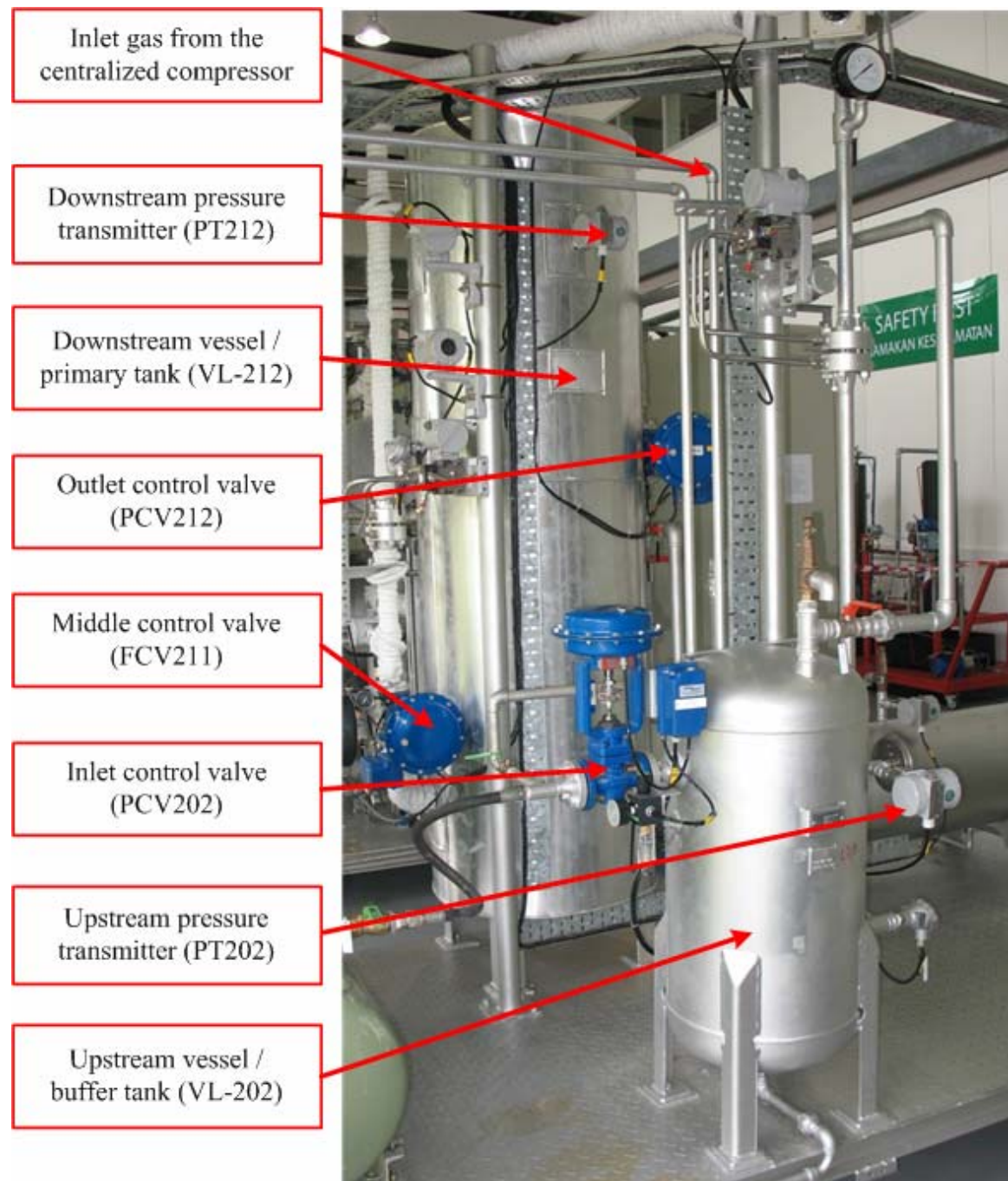


**Figure 3.2:** Schematic diagram of the gaseous pilot plant.

Some difficulties for constructing a physical model and controlling such a plant can be addressed as follow:

- a. The engineering data of the gaseous pilot plant that records the size and thickness of vessels and pipes are no more available.
- b. The plant is supplied by a centralized compressor, which is also used to supply the compressed air to the other units. Hence, it will be some fluctuations occur during the normal operation. Furthermore, the measurement of inlet pressure is not available in the local control station.
- c. The volume of the buffer tank (VL-202) is about five times smaller than the volume of primary tank (VL-212).
- d. The measurements of inlet and outlet flowrates are not available.

Considering these conditions, the empirical modeling of is preferred instead of physical modeling.



**Figure 3.3:** Photograph of the gaseous pilot plant.

### 3.4.2 Analysis of plant dynamics

Before the empirical modeling of the gaseous pilot plant using system identification technique is performed, the analysis of plant dynamics needs to be conducted. This analysis is aimed to get a priori knowledge about the interactions of variables inside the plant and employs some linearizations.

It is assumed that the density of the gas, in this case compressed air, is constant, so that the equation (2.25) can be used. Let  $F_1$  is the inlet volumetric flowrate,  $F_2$  is the volumetric flowrate from upstream vessel to downstream vessel, and  $F_3$  is the outlet volumetric flowrate. The mass balance in the upstream vessel can then be written as:

$$\frac{V_1}{\beta} \frac{dP_1}{dt} = F_1 - F_2. \quad (3.8)$$

It is also assumed that the isothermal bulk modulus  $\beta$  is constant, so that  $V_1 / \beta$  remains constant.

The pressure in the source  $P_s$  is maintained to be constant at 7 barg, even though in the real application there will be some fluctuations. So, the volumetric flowrate  $F_1$  can be written as:

$$F_1 = C_v f(u_1) \sqrt{\frac{P_s - P_1}{s.g.}}. \quad (3.9)$$

with  $u_1$  is the valve opening of the input control valve (PCV202).

It is assumed that the linear valves are used in the plant. The function of flowrate is linearized with the pressure difference. Marlin [40] points out that for the linearized system, the relation between flowrate and pressure difference in (3.9) can be rewritten as

$$F_1 = K_1 u_1 (P_s - P_1), \quad (3.10)$$

with  $K_1$  is a constant coefficient that obtained from the valve coefficient, square-root of specific gravity of compressed air, and the linearization coefficient.

Hence, for  $F_2$ , the relation becomes

$$F_2 = K_2 u_2 (P_1 - P_2), \quad (3.11)$$

with  $u_2$  is the valve opening of the middle control valve (FCV211), and for  $F_3$ :

$$F_3 = K_3 u_3 (P_2 - P_{amb}). \quad (3.12)$$

with  $K_2$  and  $K_3$  are constant coefficients, which are obtained from the valve coefficient, square-root of specific gravity of compressed air, and the linearization coefficient.  $P_{amb}$  is the ambient pressure of environment, and  $u_3$  is the valve opening of the outlet control valve (PCV212).

Then the dynamics of upstream vessel can be written as

$$\begin{aligned} \frac{V_1}{\beta} \frac{dP_1}{dt} &= K_1 u_1 (P_s - P_1) - K_2 u_2 (P_1 - P_2) \\ &= K_1 u_1 P_s - (K_1 u_1 + K_2 u_2) P_1 + K_2 u_2 P_2 . \end{aligned} \quad (3.13)$$

Taking the Laplace transform, it becomes:

$$\begin{aligned} P_1(s) &= \frac{K_1 P_s}{\left( \frac{V_1}{\beta} s + (K_1 u_1(s) + K_2 u_2(s)) \right)} u_1(s) \\ &+ \frac{K_2}{\left( \frac{V_1}{\beta} s + (K_1 u_1(s) + K_2 u_2(s)) \right)} u_2(s) P_2(s) . \end{aligned} \quad (3.14)$$

Again, the further assumption will be made. It is assumed that the dynamics of pressure balance can be approached using 1<sup>st</sup>-order transfer functions from the respective inputs, with time-invariant parameters.

The opening valves  $u_1$ ,  $u_2$ , and  $u_3$  have the following range (in percentage opening):

$$0 \leq u_{1,2,3} \leq 100 . \quad (3.15)$$

To obtain the time-invariant parameters of 1<sup>st</sup>-order transfer function, it is assumed that the valves only vary at small range of their operating conditions. So that the following assumption is made:

$$\frac{1}{K_1 u_1(s) + K_2 u_2(s)} \approx K_{1,v,ss} = \text{constant} . \quad (3.16)$$

Then the 1<sup>st</sup>-order transfer function of pressure dynamics in upstream vessel can be written as

$$P_1(s) = \frac{K_{1,v,ss} K_1 P_s}{\left( \frac{K_{1,v,ss} V_1}{\beta} s + 1 \right)} u_1(s) + \frac{K_{1,v,ss} K_2}{\left( \frac{K_{1,v,ss} V_1}{\beta} s + 1 \right)} P_2(s) u_2(s) . \quad (3.17)$$

For downstream vessel, its dynamics can be written as

$$\begin{aligned} \frac{V_2}{\beta} \frac{dP_2}{dt} &= K_2 u_2 (P_1 - P_2) - K_3 u_3 (P_2 - P_{amb}) \\ &= K_2 u_2 P_1 - (K_2 u_2 + K_3 u_3) P_2 + K_3 u_3 P_{amb} . \end{aligned} \quad (3.18)$$

Taking the Laplace transform, it becomes:

$$P_2(s) = \frac{K_2}{\left(\frac{V_2}{\beta}s + (K_2u_2(s) + K_3u_3(s))\right)} P_1(s)u_2(s) + \frac{K_3P_{amb}}{\left(\frac{V_2}{\beta}s + (K_2u_2(s) + K_3u_3(s))\right)} u_3(s). \quad (3.19)$$

Again, the further assumption will be made. It is assumed that the dynamics of pressure balance can be approached using 1<sup>st</sup>-order transfer functions from the respective inputs, with time-invariant parameters.

The same assumption is made from the downstream pressure, which is

$$\frac{1}{K_2u_2(s) + K_3u_3(s)} \approx K_{2,v,ss} = \text{constant}. \quad (3.20)$$

Then the 1<sup>st</sup>-order transfer function of pressure dynamics in downstream vessel can be written as

$$P_2(s) = \frac{K_{2,v,ss}K_2}{\left(\frac{K_{2,v,ss}V_2}{\beta}s + 1\right)} P_1(s)u_2(s) + \frac{K_{2,v,ss}K_2P_{amb}}{\left(\frac{K_{2,v,ss}V_2}{\beta}s + 1\right)} u_3(s). \quad (3.21)$$

The pressures inside the vessels will be in a steady-state if

$$\frac{dP_1}{dt} = 0, \quad (3.22)$$

$$\frac{dP_2}{dt} = 0. \quad (3.23)$$

Assuming that the valves are linear, the volumetric flowrate of the gas passes through the  $i^{\text{th}}$  control valve can be written as:

$$F_i = K_{ss,i}u_i\sqrt{\Delta P}, \quad (3.24)$$

with  $K_{ss,i} = C_{v,i} / \sqrt{s.g.}$  and  $u_i$  is the of valve opening for linear valves, and  $i = 1, 2, 3$ .

The steady-state pressure in the upstream and downstream vessel can be obtained by solving the following equations:

$$K_{ss,1}u_1\sqrt{P_s - P_1} = K_{ss,2}u_2\sqrt{P_1 - P_2}, \quad (3.25)$$

$$K_{ss,2}u_2\sqrt{P_1 - P_2} = K_{ss,3}u_3\sqrt{P_2 - P_{amb}} . \quad (3.26)$$

From (3.17) until (3.26) it can be implied that

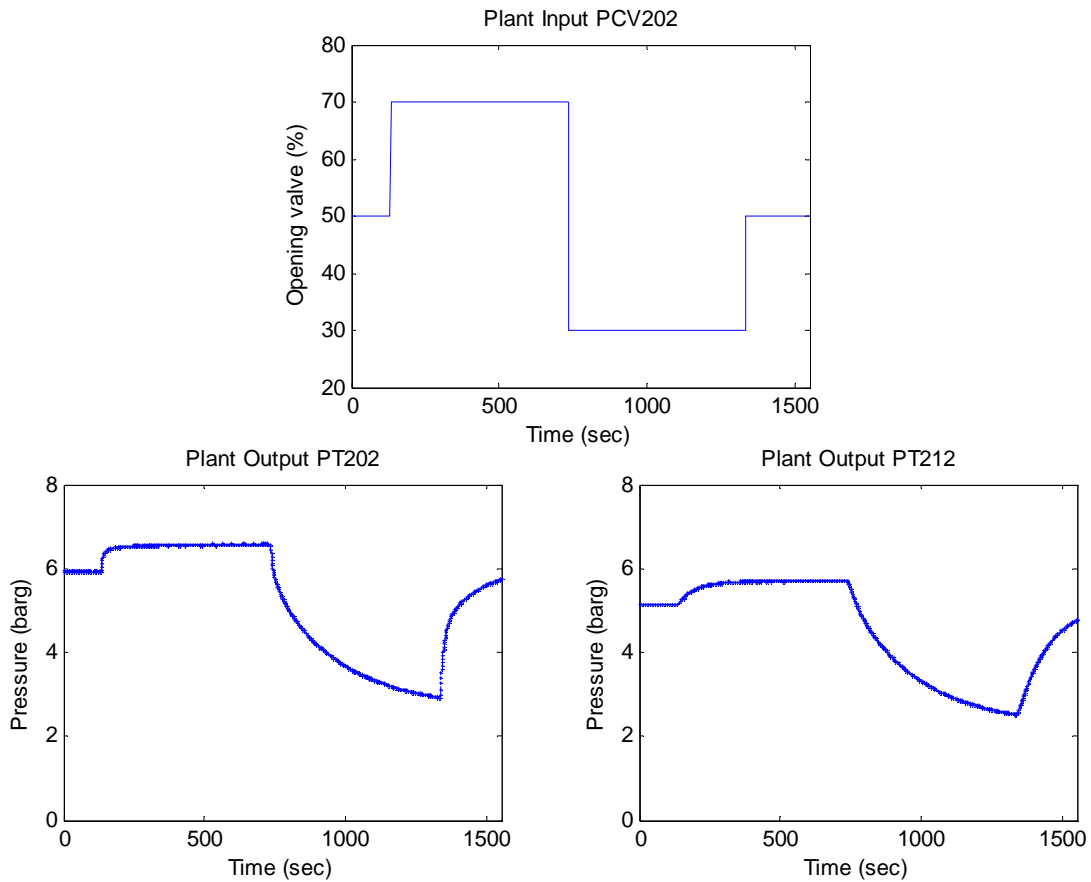
$$P_1(t) = f(u_1, u_2, u_3, P_2, t) , \text{ and} \quad (3.27)$$

$$P_2(t) = f(u_1, u_2, u_3, P_1, t) . \quad (3.28)$$

### 3.4.3 Analysis of plant responses

The plant dynamics as the responses from the three external inputs: inlet control valve (PCV202), middle control valve (FCV211), and outlet control valve (PCV212) are partially analyzed as follows:

#### 3.4.3.1 The responses from changes in the inlet control valve



**Figure 3.4:** The step changes in inlet control valve (top), the upstream response (left-bottom), and the downstream response (right-bottom) of the gaseous pilot plant, while the two others control valve are kept to be constant at 50% of valve openings.

Figure 3.4 shows the upstream and the downstream responses from step changes in the inlet control valve (PCV202), while the two others control valve (FCV211 and PCV212) are kept to be constant at 50% of valve openings. It can be analyzed from the figure that the response of PT202 in this condition has faster time constant compare to the response of PT212. When the inlet control valve is increased from 50% to 70%, the responses of both measurements, the PT202 and PT212, exhibit a first order system that quickly reached their steady-state condition. The upstream response reaches its steady-state condition at 6.5 barg from the initial condition: 6 barg, and the downstream response reached its steady-state condition at 5.7 barg from the initial condition: 5.1 barg. However, when the inlet control valve is reduced from 70% to 30%, both outputs continuously ramp down but do not reach their steady-state. Based on the definition of a linearity given in section 2.1, a linear system has to satisfy the following:

*Theorem 3.1:*

*The response of a linear model is proportional to the input. That is, if for a given input,  $u(t)$ , the response is  $y(t)$ , then for an input  $\alpha u(t)$  the response is  $\alpha y(t)$ ,  $\alpha \in \mathbb{R}$ .*

Let all the input and output variables are normalized to their initial conditions. Let  $u_1$ ,  $u_2$ , and  $u_3$  represent the valve opening of PCV202, FCV211, and PCV212 respectively, after being normalized to their initial conditions, which are 50% of valve openings. Let  $y_1$  and  $y_2$  represent the upstream and downstream pressures (PT202 and PT212) respectively, after being normalized to their initial conditions, which are 6 barg for PT202 and 5.1 barg for PT212. Let the plant inputs,  $u(t)$ , is defined as

$$u(t) = u_1(t) + u_2(t) + u_3(t). \quad (3.29)$$

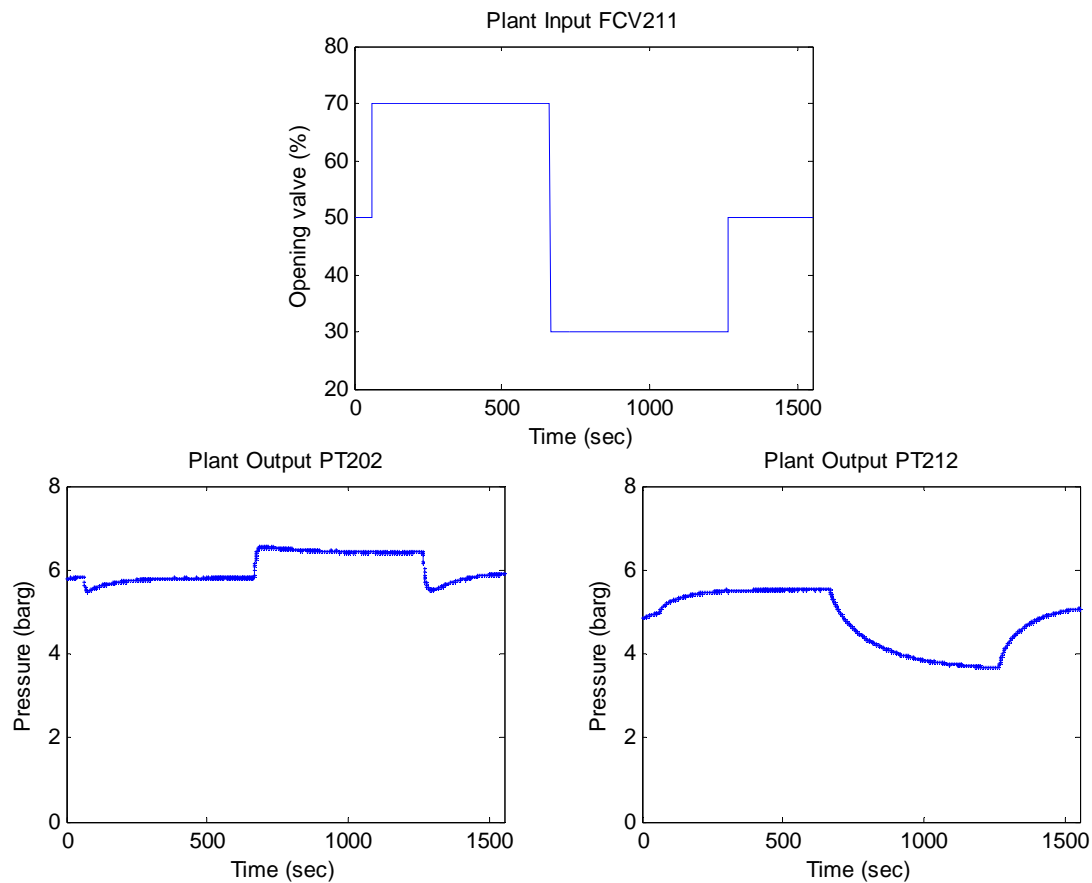
Since FCV211 and PCV212 are kept to be constant at their initial conditions, then  $u_2(t)$  and  $u_3(t)$  in (3.29) are equal to zero. Let  $\alpha$  equal to -1 and the upstream and downstream responses are analyzed separately.

For  $t \rightarrow \infty$ ,  $y_1$  reach its steady state condition at 0.5 barg (after normalization) when  $u(t) = u_1(t) = 20\%$  of valve opening. If the linearity satisfied, based on Theorem 3.1,  $y_1$  should reach its steady state condition at -0.5 barg (after normalization) when the input is  $\alpha u(t) = \alpha u_1(t) = -20\%$  of valve opening. However, according to the response data in

Figure 3.4, this condition is not satisfied. The similar thing happens for  $y_2$ . Hence, it can be said that the upstream and downstream responses of the gaseous pilot plant are nonlinear from changes in the inlet control valve (PCV202).

Suppose that the linear condition is expected, the steady-state plant gains from the inlet control valve are: 0.025 (barg/(% valve opening)) for upstream, and 0.03 (barg/(% valve opening)) for downstream. Even if the linearity responses can be obtained, this condition gives small linearity ranges compare to the desired operating ranges. So that, considering the desired operating ranges, it can be said that the nonlinearities of plant responses from the inlet control valve (PCV202) cannot be avoided.

### 3.4.3.2 The responses from changes in the middle control valve

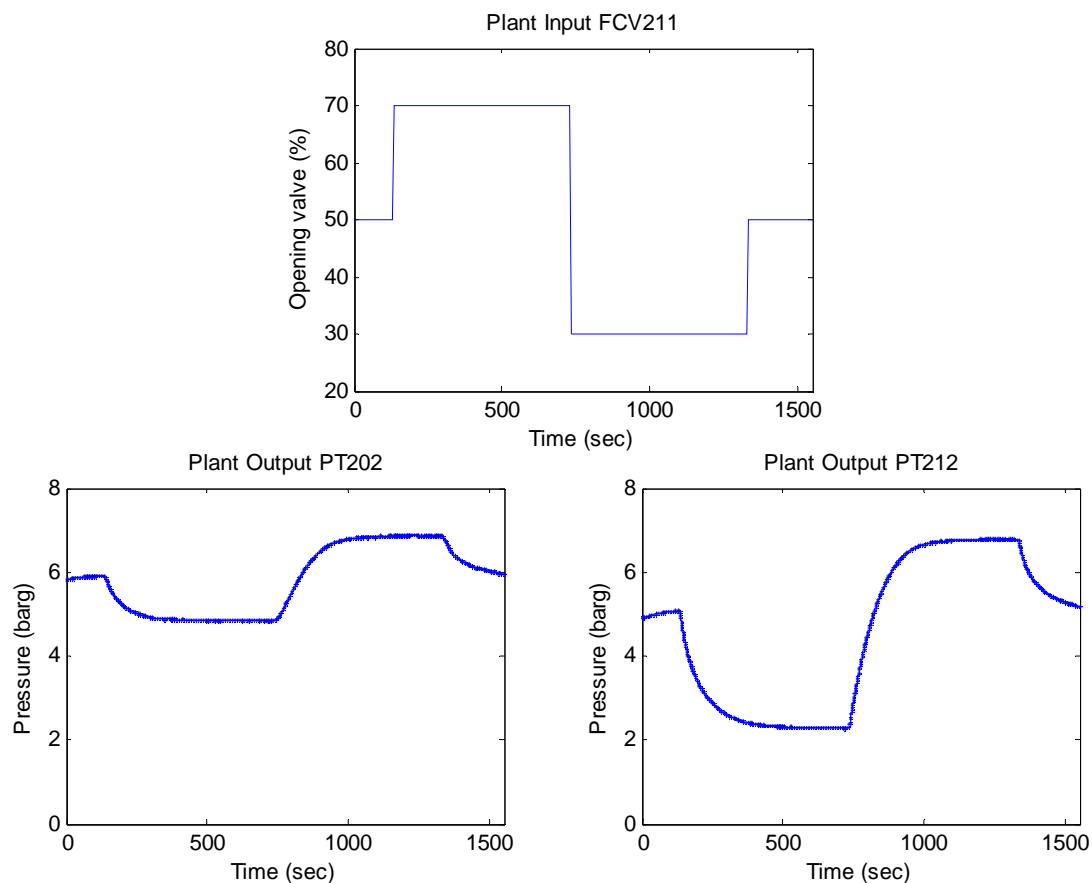


**Figure 3.5:** The step changes in middle control valve (top), the upstream response (left-bottom), and the downstream response (right-bottom) of the gaseous pilot plant, while the two others control valve are kept to be constant at 50% of valve openings.



The upstream and the downstream responses of the gaseous pilot plant from step changes in the middle control valve (FCV211) are shown in Figure 3.5. The responses of the upstream pressure shows the inverse responses, which means the step responses changes initially in the opposite direction from the final steady-state [40]. Unlike the upstream response, the pressure in the downstream shows a similar response with the response from step changes in PCV202, but having a different gain and time constant. For both the upstream and downstream pressures, the responses from step changes in the middle control valve FCV211 exhibit nonlinearity. However, the steady-state gains of the plant from the changes in FCV211 are very small, especially for upstream response. This also implies that the middle control valve may give a poor controllability when it is used as the manipulated variable to control the upstream or the downstream pressures [83].

### 3.4.3.3 The responses from changes in the outlet control valve

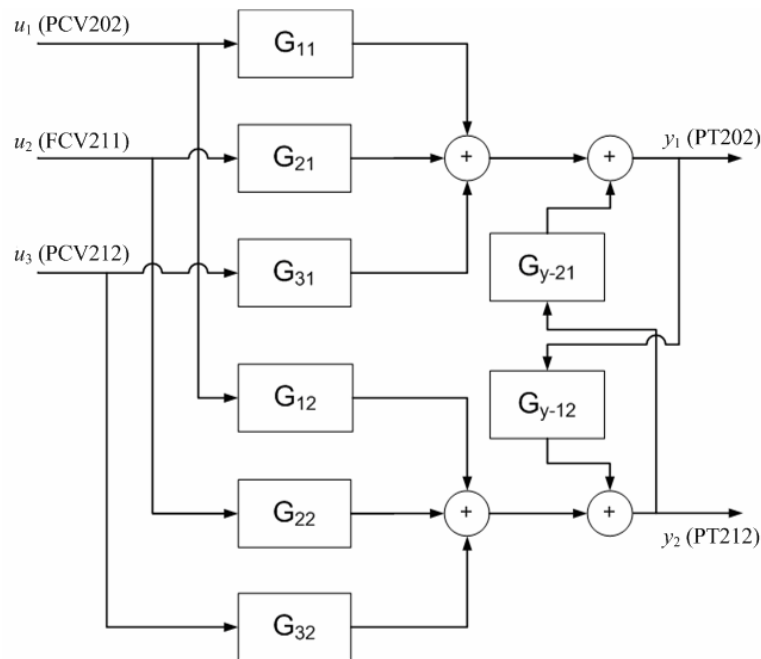


**Figure 3.6:** The step changes in outlet control valve (top), the upstream response (left-bottom), and the downstream response (right-bottom) of the gaseous pilot plant, while the two others control valve are kept to be constant at 50% of valve openings.

The responses from step changes in the outlet control valve (PCV212) for the upstream and the downstream pressures are shown in Figure 3.6. From the figure, it seems that both responses show the almost linear responses. Both plant outputs return to their initial conditions after the input variable (opening valve of PCV212) is brought back to its initial condition (50% of valve opening). In this condition, the output variables are also self-regulated, which means the output variables tend to settle to steady-state after the input variable has reached a constant value. Either upstream or downstream pressure also has almost the same steady-state gain when the control valve PCV212 moves up or down. The upstream and the downstream responses also show a big steady-state plant gain, which denotes that the outlet control valve may produce a good controllability when it is used as the manipulated variable.

### 3.4.4 Models structures

From the analyses of plant dynamics and their responses of the changes on each control valve, and considering the interaction from one to another output variable, the dynamics of the gaseous pilot plant is then illustrated as in Figure 3.7, with  $G_{ij}$  represents the transfer function of  $i^{\text{th}}$ -input to  $j^{\text{th}}$ -output while  $G_{y-ij}$  represents the transfer function of  $i^{\text{th}}$ -output to  $j^{\text{th}}$ -output.



**Figure 3.7:** The dynamics of gaseous pilot plant.

According to the representation given in Figure 3.7, the dynamics of the plant can be written as

$$y_1 = G_{11} \times u_1 + G_{21} \times u_2 + G_{31} \times u_3 + G_{y-21} \times y_2, \quad (3.30)$$

and

$$y_2 = G_{12} \times u_1 + G_{22} \times u_2 + G_{32} \times u_3 + G_{y-12} \times y_1. \quad (3.31)$$

There are two ways to develop the linear model from input-output data which have the relation as given above. These ways can be explained as follows:

- a. The first is by using a MIMO structure which represents the complete relation in (3.30) and (3.31). In more compact form, (3.30) can be rewritten as

$$y_1 = \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + G_{y-21} \times y_2, \quad (3.32)$$

and (3.31) can be rewritten as

$$y_2 = \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + G_{y-12} \times y_1. \quad (3.33)$$

By substituting (3.32) into (3.33), the following will be obtained:

$$y_1 = \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + G_{y-21} \times \left[ \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + G_{y-12} \times y_1 \right]; \quad (3.34)$$

$$(1 - G_{y-21} \times G_{y-12}) \times y_1 = \left( \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} + G_{y-21} \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} \right) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}; \quad (3.35)$$

$$y_1 = (1 - G_{y-21} \times G_{y-12})^{-1} \left( \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} + G_{y-21} \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} \right) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (3.36)$$

In the same way, (3.33) can be rewritten as

$$y_2 = (1 - G_{y-21} \times G_{y-12})^{-1} \left( \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} + G_{y-12} \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} \right) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (3.37)$$

(3.36) and (3.37) can be combined into a matrix notation, so that it becomes

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} (1 - G_{y-21} \times G_{y-12})^{-1} \left( \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} + G_{y-21} \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} \right) \\ (1 - G_{y-21} \times G_{y-12})^{-1} \left( \begin{bmatrix} G_{12} & G_{22} & G_{32} \end{bmatrix} + G_{y-12} \begin{bmatrix} G_{11} & G_{21} & G_{31} \end{bmatrix} \right) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (3.38)$$

- b. The model structure in (3.38) implies that suppose the linear parameters  $G_{ij}$  and  $G_{y-ij}$  are obtained, then the variable  $y_1$  and  $y_2$  can be estimated or predicted from a given set of values:  $u_1$ ,  $u_2$ , and  $u_3$ .

Suppose that  $y_1$  is to be predicted and there are no changes in  $u_2$ , and  $u_3$ , then  $y_1$  is linearly depended on the  $u_1$ . Nevertheless, the response of  $y_1$  from the changes in  $u_1$  (PCV202), as shown in Figure 3.4, is nonlinear. But, it also can be implied from the responses plot in Figure 3.4 that the upstream pressure,  $y_1$ , and the downstream pressure,  $y_2$ , have a good correlation. Hence, if the downstream pressure,  $y_2$ , is taken as one of the measured inputs,  $y_1$  can seemly be approached as the linear combination of the function of the external inputs ( $u$ ) and  $y_2$ .

Considering that involving the measurement of  $y_2$  can be much useful for estimating  $y_1$  from the model, the relation in (3.30) can also be reformed as

$$y_1 = \begin{bmatrix} G_{11} & G_{21} & G_{31} & G_{y-21} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ y_2 \end{bmatrix}. \quad (3.39)$$

And for  $y_2$ , it becomes

$$y_2 = \begin{bmatrix} G_{12} & G_{22} & G_{32} & G_{y-12} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ y_1 \end{bmatrix}. \quad (3.40)$$

Unlike (3.38), the two model structures in (3.39) and (3.40) cannot be simply combined in a single matrix notation, since one output becomes the input of another model. Since having a variable as the input and at the same time as the output is not

allowed in system identification procedures, then the model parameters in (3.39) and (3.40) have to be estimated separately.

Suppose all the linear model parameters  $G_{ij}$  and  $G_{y-ij}$  in (3.38), (3.39), and (3.40) are successfully obtained using system identification technique, predicting  $y_1$  using model structure in (3.38) only requires the measurement data of  $u_1$ ,  $u_2$ , and  $u_3$ , while when model structure in (3.39) is used, predicting  $y_1$  also requires the measurement data of  $y_2$ , and vice versa for the prediction of  $y_2$ .

### 3.5 System Identification of Gaseous Pilot Plant

In this section, three different approaches are used to perform the empirical modeling of the gaseous pilot plant using subspace system identification methods. These approaches can be briefly explained as follows:

- a. The first approach is using (3.38) as the model structure. The input signals are step signals, which will be applied to the three control valve sequentially. The resulted model is named Model 1.
- b. The second approach still using (3.38) as the model structure, while the pseudo-random binary sequence (PRBS) signals are selected the input signal. The design of three PRBS signals is carried out independently and will be applied to the three control valves simultaneously. For this purpose, the approximate first-order plus delay time (FOPDT) transfer functions of the plant are considered and the information of the signal and noise statistics are analyzed. This model is named Model 2.
- c. The third approach is by using (3.39) and (3.40) as the models structure. The step signals are chosen as the input signals, and the identification processes are performed as two separate MISO identifications. The MISO models are further combined to a single MIMO state-space model. The resulted model is named Model 3.

The experimental setup that would be used to perform system identification processes of the gaseous pilot plant in this research includes the following:

a. Hardware packages

To design the plant testing and data acquisition configurations, a general purpose personal computer (PC) named host PC would be used. As the real-time processor, a kind of Pentium-based industrial computer called xPC TargetBox would be employed. A Diamond-MM-32-AT analog-to-digital (A/D) and digital-to-analog (D/A) converter would be used as the data acquisition module. This module includes 16-bit A/D converter as the analog input module and 12-bit D/A converter as the analog output module.

b. Software packages

MATLAB and Simulink softwares would be used to design the plant testing and data acquisition applications. To build the real-time application, Real-Time Workshop (RTW) and xPC Target would be employed. A MATLAB script would be created to monitors and logs the data in every one second and then automatically stores the logged data into a spreadsheet file when the application finished.

To estimate the model parameters, the System Identification Toolbox from MATLAB would be used in this work. This toolbox provides a subspace system identification tool using N4SID algorithm that can be called using *n4sid* syntax.

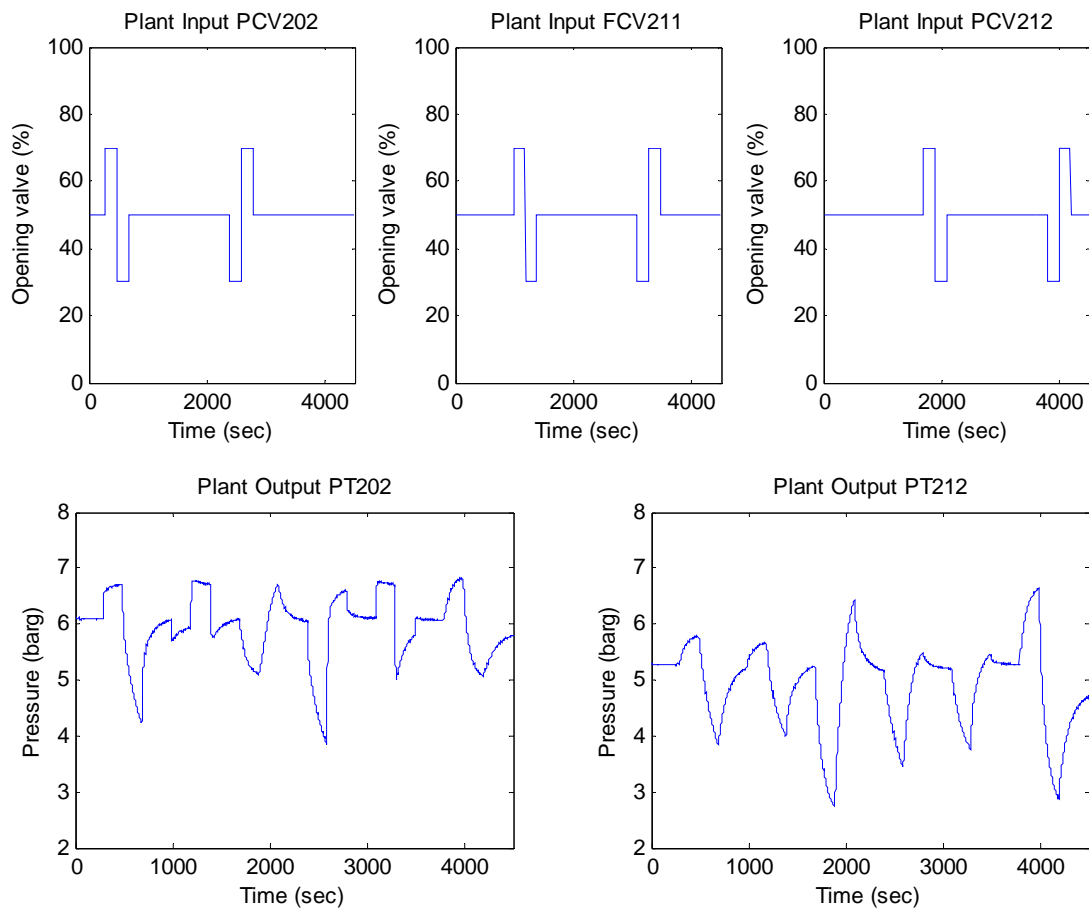
### 3.5.1 Model 1

#### 3.5.1.1 Plant testing

Considering the operating conditions of the plant, the step input signal is designed to have the amplitude range from 30% to 70% of valve opening. The sampling time used in the experiment is one second, and the total recorded data is 4500. The identification process would be carried out off-line using the first half of the data set (2250), whereas the remaining data would be used for model validation.

Figure 3.8 shows the input-output data used in the development of Model 1. Initially, the three control valves are maintained at 50% of valve opening until both outputs are considerably in steady-state conditions. The step signal is then applied to the inlet control valve, while the other two control valves are maintained to be constants. The same procedure is then applied to the middle control valve and the outlet control valve. The

step signals which are transmitted to these three control valves have the same length and amplitude.

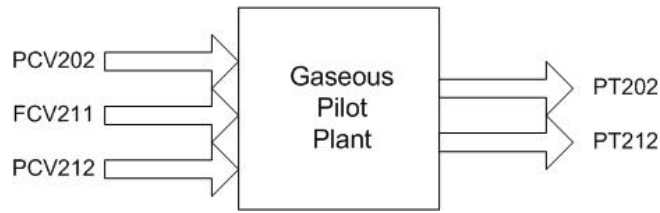


**Figure 3.8:** Input signals and plant responses for the development of Model 1.

It may be noted that the open-loop identification procedure would be conducted and the purely deterministic case would be considered.

### 3.5.1.2 *Model structure and variable selection*

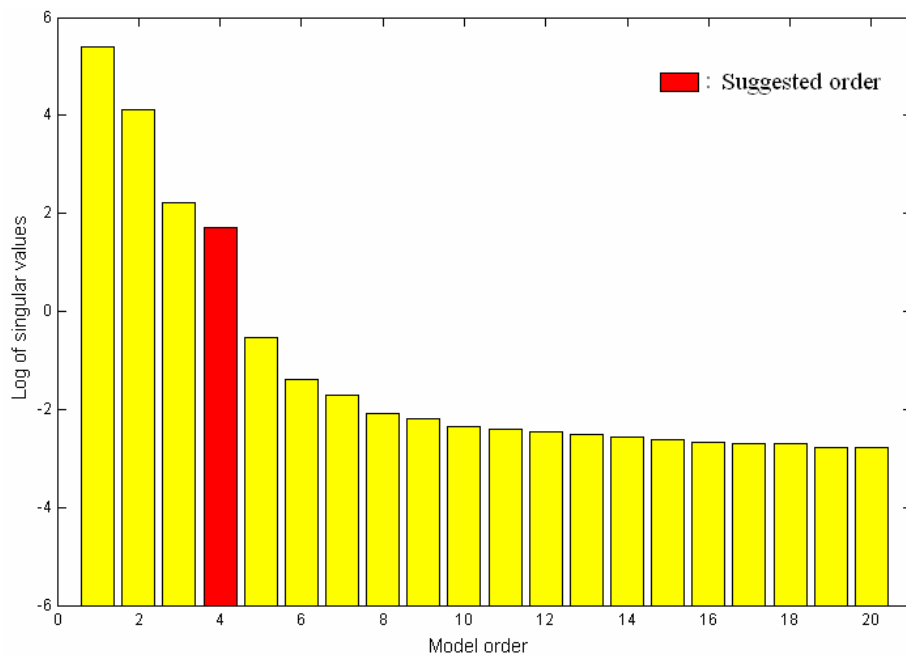
Since (3.38) is used as the model structure, the state-space model that will be constructed in this sub-section has three inputs and two outputs. The first output is represented by the pressure inside buffer tank (upstream pressure) that is indicated by the pressure transmitter PT202, while the second output is represented by the pressure inside primary tank (downstream pressure) which is indicated by the pressure transmitter PT212. The plant inputs are represented by the opening of the three control valves: PCV202, FCV211, and PCV212, respectively. This model structure is illustrated in Figure 3.9.



**Figure 3.9:** The schematic diagram of the first model structure.

### 3.5.1.3 Order estimation

There is an extensive literature on order estimation algorithms for linear, dynamical, state-space systems that includes [84-86]. Nevertheless, Sotomayor et al. [75] point out that there exist only few references dealing with the estimation of the order in the context of subspace identification methods. In this research, three approaches are used for estimating the order of state-space model, i.e., by examining the singular value decompositions of N4SID algorithm, the Hankel singular values, and the simulation errors.

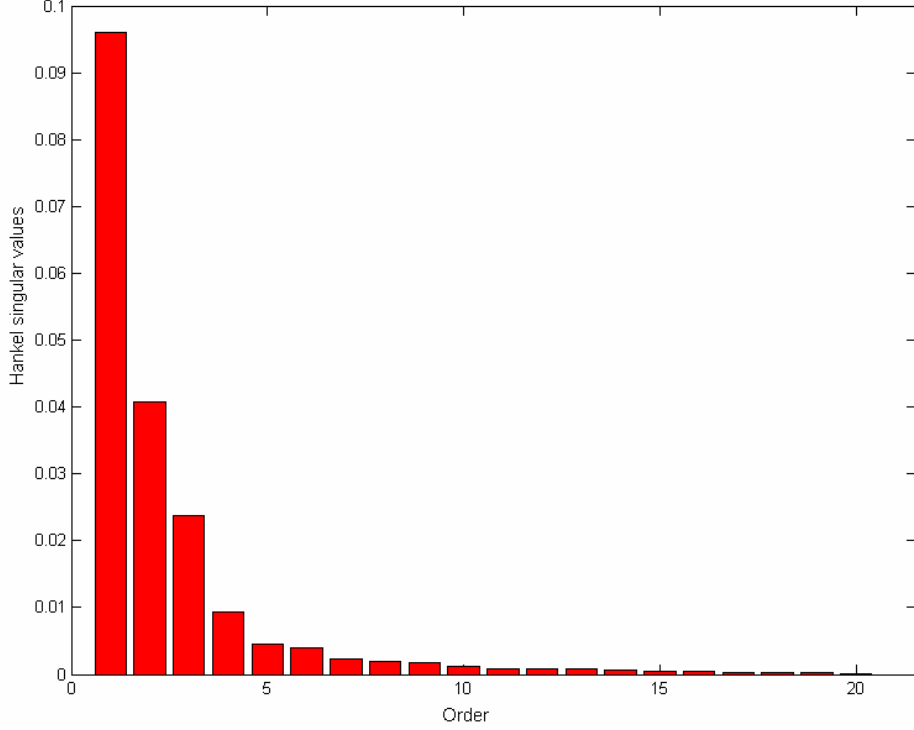


**Figure 3.10:** Model singular values plot with the suggested model order (red) given by N4SID algorithm.

As outlined in the section 4.2, the N4SID algorithm performs the singular value decomposition to estimate the order  $n$  of the state-space model. The first consideration to



determine the model order in this work is by examining the results of such decomposition. Figure 3.10 shows the results of singular value decomposition performed by N4SID algorithm, which suggests the system order is  $n = 4$ .



**Figure 3.11:** Hankel singular values plot.

The second approach to determine the model order in this work is by examining the Hankel singular values that define the “energy” of each state in the system. This method is adopted from model reduction techniques [87, 88] with a priori assumption that greater model order gives better estimation. The idea is to observe the states that much influence the overall system, since keeping larger energy states of the system preserves most of its characteristics in terms of stability, frequency, and time responses [87, 88].

For given a stable state-space system, with  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are its coefficient matrices, the Hankel singular values,  $\sigma_H$ , are defined as [87-90]

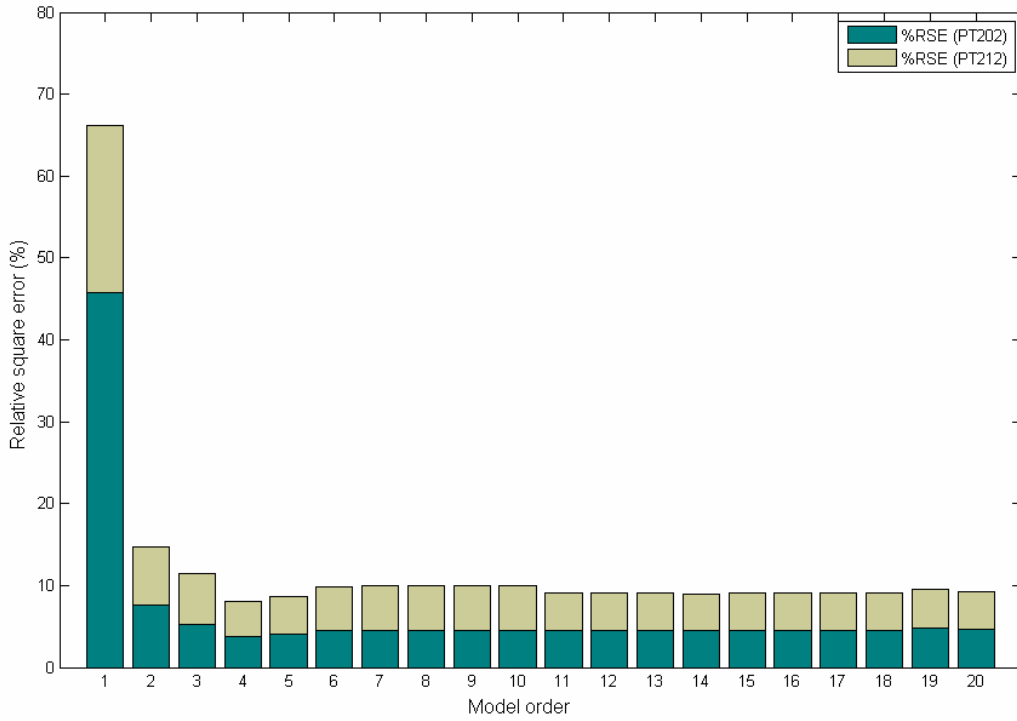
$$\sigma_H = \sqrt{\lambda_i(\mathbf{PQ})}, \quad (3.41)$$

where  $\lambda_i(\mathbf{PQ})$  denotes the  $i$ -th eigenvalue of  $\mathbf{PQ}$ , and  $\mathbf{P}$ ,  $\mathbf{Q}$  are controllability and observability grammians determined by the discrete-time Lyapunov equations

$$\mathbf{A}\mathbf{P}\mathbf{A}^T - \mathbf{P} + \mathbf{B}\mathbf{B}^T = 0, \quad (3.42)$$

$$\mathbf{A}^T \mathbf{Q} \mathbf{A} - \mathbf{Q} + \mathbf{C}^T \mathbf{C} = 0. \quad (3.43)$$

With an assumption that a greater model order may give a better estimation result, then the 20<sup>th</sup>-order state-space model is first constructed from input-output data using N4SID algorithm. The obtained coefficient matrices of this model (**A**, **B**, **C**, and **D**) are used to calculate the Hankel singular values of its states, hence the contribution of each state to overall system characteristics could be analyzed. The plot of the Hankel singular values of such a system is given in Figure 3.11. As shown in the figure, the first four orders of the system store the most its characteristics. However, the greater orders such as 5 or 6 are also possible choices.



**Figure 3.12:** Relative square error (RSE) plot of Model 1.

The third approach is by choosing the order  $n$  that minimize the simulation errors. According to Bastogne et al. [81] and Sotomayor et al. [75], a more practical procedure for order estimation is to choose the value  $n$  that minimize simulation errors. Figure 3.12 shows relative square errors of the system generated using N4SID algorithm, from order  $n = 1$  to 20. Among the 20 variations of the model order, it is found that the 4<sup>th</sup>-order system gives the minimum simulation errors.

Considering the suggested order by N4SID algorithm from the singular value decomposition, the inspection of the Hankel singular values, and the observation of the relative estimation errors indexes, the state-space model of gaseous pilot plant for Model 1 is chosen to be of 4<sup>th</sup>-order.

#### 3.5.1.4 Identification result and model validation

To obtain the parameters of Model 1, in this case the elements of its coefficient matrices, the input-output data shown in Figure 3.8 are first divided into two sets of data. The first set of data, which are contain the input signals and the plant responses during the first 2250 sampling time, are used to estimate the model parameters using N4SID algorithm. The pre-treatment of the input-output data such as noise filtering and outlier removal are not conducted, since the data seem to have a big signal-to-noise ratio (*S/N ratio*) and no outliers are detected.

From the results of parameter identification and estimation using N4SID method, the Model 1 has state coefficient matrices for the discrete linear time-invariant with one second sampling time as shown in (3.44) until (3.47). The poles of plant model (the eigenvalues of **A** matrix), which are shown in Figure 3.13, are inside the unit circle denote that the system is stable. The poles closer to the unit circle are related to the slower system dynamics. The poles close to 1, shown that the data set seems to contain a phenomenon known as “co-integration” in econometrics [75]. Based on this observation, it is possible to obtain models which produce a one-step-ahead prediction error much smaller [75].

$$\mathbf{A} = \begin{bmatrix} 0.9959 & -0.0031 & -0.0069 & 0.0017 \\ -0.0021 & 0.9847 & 0.0250 & 0.0246 \\ 0.0017 & 0.0216 & 0.9374 & -0.0463 \\ -0.0095 & -0.0040 & -0.0157 & 0.9572 \end{bmatrix}; \quad (3.44)$$

$$\mathbf{B} = \begin{bmatrix} 0.00003 & -0.00001 & -0.00002 \\ -0.00018 & 0.00020 & -0.00006 \\ 0.00039 & -0.00041 & 0.00006 \\ 0.00018 & -0.00012 & 0.00001 \end{bmatrix}; \quad (3.45)$$

$$\mathbf{C} = \begin{bmatrix} 28.405 & -12.521 & -0.110 & -0.096 \\ 34.682 & 5.394 & 0.011 & -0.013 \end{bmatrix}; \quad (3.46)$$

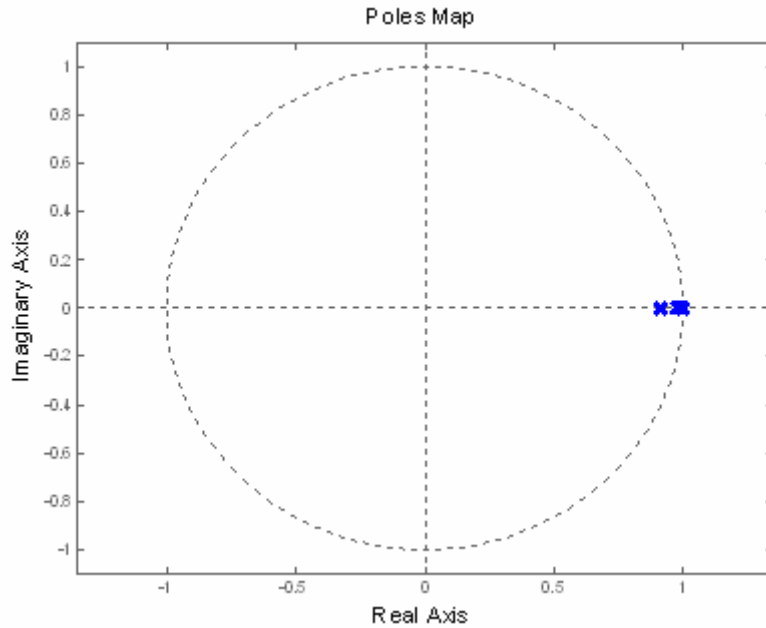
$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.47)$$

The comparison of the outputs generated by Model 1 and the real outputs of the plant used as the validation data (the input signals and the plant responses during the last 2250 sampling time) are shown in Figure 3.14. The solid line represents the real measured outputs of the plant, while the dotted line represents the simulation output produced by Model 1. The simulation starts initially with an estimated state.

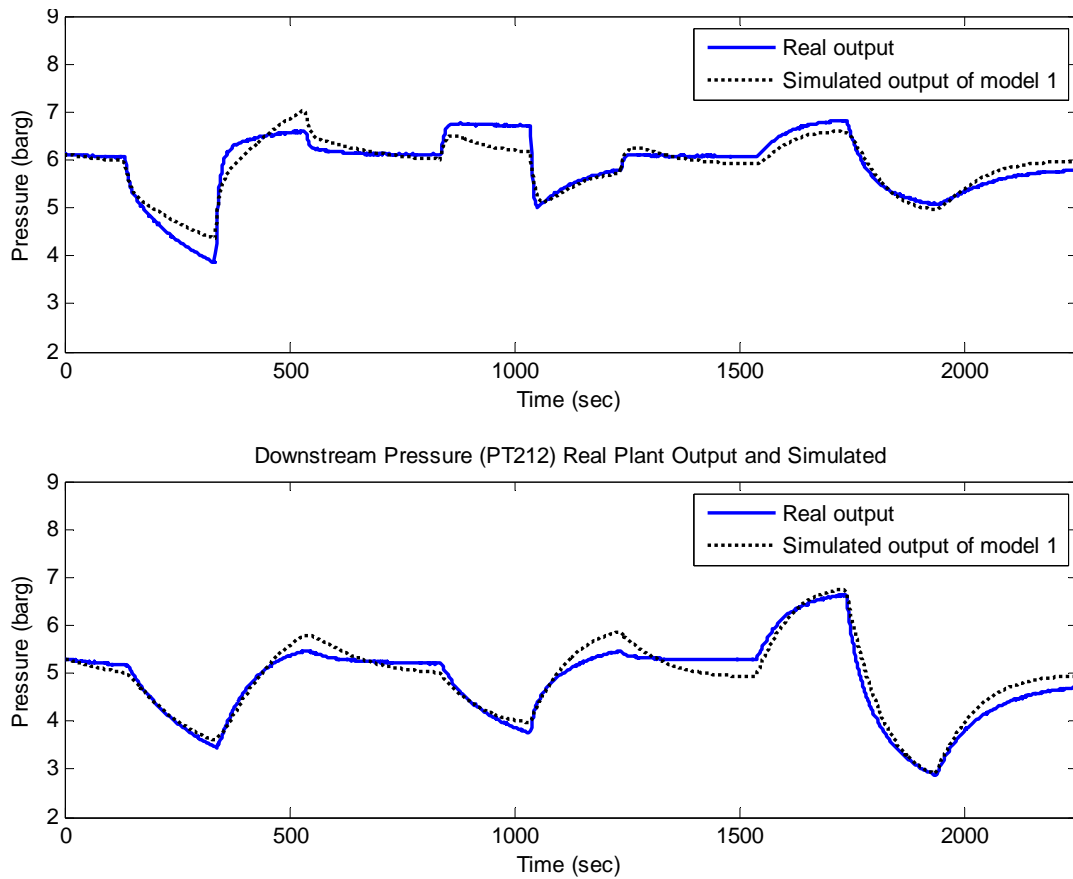
To examine the goodness of fit of the model, the percentage of the output that the model reproduces, to be called the best fit criterion, is used. For  $N$  number data, The best fit criterion of a model is calculated as [91]:

$$\text{Best Fit} = \left( 1 - \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{\sum_{i=1}^N |y_i - \bar{y}|} \right) \times 100\%, \quad (3.48)$$

with  $\bar{y}$  is the mean of measured output. The best fit criterion of Model 1 for identification and validation data are shown in Table 3.1.



**Figure 3.13:** Location of the poles of Model 1.



**Figure 3.14:** The comparison of real output and simulated output generated by Model 1 for upstream pressure (top) and downstream pressure (bottom).

**Table 3.1:** The Performance of Model 1.

Plant output	Best fit criterion (%)	
	Identification data	Validation data
PT202 (upstream pressure)	35.78	66.41
PT212 (downstream pressure)	55.40	74.78

### 3.5.2 Model 2

Another way to model the dynamic characteristics of a process is by using frequency domain representation. Frequency response of an input-output data set,  $u(t)$  and  $y(t)$ , is given by

$$g(i\omega) = \frac{y(i\omega)}{u(i\omega)}, \quad (3.49)$$

where  $y(i\omega)$  and  $u(i\omega)$  are the Fourier transforms of the output and input sequence respectively. This equation implies that to characterize the behavior of a process over a given frequency range, the input signal must be rich (contain sufficient power or strength) in that frequency range. If  $u(i\omega)$  is small, then the corresponding excitation in  $y(i\omega)$  is also small. Hence, the small errors in  $y$  or  $u$  will cause large errors in the model at that frequency. This insight will be used in designing input signals for identification tests.

One of the commonly employed input signals is the pseudo-random binary sequence (PRBS) signal. A PRBS consists of a series of up and down step changes in the input variable. It is a deterministic, two-state binary sequence.

The PRBS signal has the following characteristics:

- a. A PRBS perturbs the system around a steady-state and does not result in significant deviations from this value. Since the test can last over a significant period, this is very important in preventing significant upsets to the production schedule during the tests.
- b. The frequency content of the PRBS can be adjusted relatively easily to meet the desired specifications on the desired frequency range.
- c. To increase accuracy of estimates, the PRBS signal can be repeated many times.

According to Brosilow and Joseph [92], detail of PRBS including its properties can be explained as follows:

- a. It is a periodic sequence with amplitude  $\pm a$  and period  $P$ . The value of the sequence changes at discrete times  $(k \times T_c)$  where  $k$  is integer and  $T_c$  is called clock tick time.
- b. In every period of length  $P$  there are  $T$  clock ticks ( $T = P/T_c$ ). There are  $T/2$  intervals over which the signal is of length 1 (one clock tick),  $T/4$  intervals over which the signal is of length 2,  $T/8$  intervals over which the signal is of length 3, and so on. This distribution effectively perturbs the plant over a wide variety of frequency ranges.  $T$  is given by  $T = 2^n - 1$ , with  $n$  is the number of shift registers used to generate the PRBS signal.

c. The power spectrum of a PRBS is given by

$$\Phi_{uu}(\omega) = \frac{\alpha^2(T+1)}{T} \left[ \frac{\sin(\omega T_c / 2)}{\omega T_c / 2} \right]^2, \quad \omega = \frac{2\pi k}{TT_c}, \quad k = 1, 2, \dots, T, \dots, \text{etc.} \quad (3.50)$$

The value of  $\alpha$  will be explained further.

### 3.5.2.1 Preliminary test

The objective of the preliminary test is to get an overview about the nature of the system being modeled. During the pre-testing phase, it is assumed that there are no faulty and out-of-calibration instruments, sensors, and/or improperly sized or installed valves [92].

The preliminary study of the plant responses is expected to give an estimate of the responses time (order of magnitude estimates) and the process gains. The data also can be used to compute the statistics of noise. This includes the standard deviation of the noise as well as a rough idea of the frequency content of the noise [92].

If it is available, the first-order plus delay time (FOPDT) transfer function of the plant can also be considered as the prior information of the plant, and be further used to design the input signals [92]. However, since there are nonlinearities in the gaseous pilot plant, it is not plausible to obtain the true FOPDT transfer functions of the plant outputs from the respective plant inputs (control valves). In this work, the prior information of plant are chosen as the FOPDT transfer functions that best fit to the step responses data which is used in the development of Model 1.

The empirical estimations to determine the process gains, time constants, and delay times of the gaseous pilot plant from the respective inputs are conducted in batches form for each plant outputs using the prediction error method [91]. The data of plant outputs are also used to determine the statistical properties of signals and noises, in this case their standard deviations, which be further used to calculate the signal-to-noise ( $S/N$ ) ratio. Based on the empirical estimation results, the first-order plus delay time transfer functions of upstream and downstream plant outputs, PT202 and PT212 respectively, are shown in Table 3.2. The statistical properties of signals and noises of both plant outputs are shown in Table 3.3.

**Table 3.2:** The FOPDT transfer functions of plant outputs from the respective inputs

Plant output	Plant inputs		
	PCV202	FCV211	PCV212
PT202	$\frac{0.054}{38s+1}$	$\frac{-0.029e^{-s}}{2s+1}$	$\frac{-0.06}{102s+1}$
PT212	$\frac{0.046e^{-12s}}{61s+1}$	$\frac{0.068}{140s+1}$	$\frac{-0.109}{74s+1}$

**Table 3.3:** The standard statistical properties of signals and noises

Plant output	Standard deviation		Signal-to-noise (S/N) ratio
	Signal	Noise	
PT202	0.5990	0.0068	7760
PT212	0.7812	0.0065	14444

### 3.5.2.2 Input signals design

The key information that is needed to design a PRBS signal is the specification of the frequency range of interest. Suppose that the process has an open-loop dominant time constant equals to  $\tau$ , the suitable frequency range of interest in the control will be

$$\frac{1}{\beta\tau} < \omega < \frac{\alpha}{\tau}. \quad (3.51)$$

As suggested in [92, 93] the values are 3 for  $\beta$  and 2 for  $\alpha$ . This assumes that a control system will be designed so that the plant would respond at least twice as fast in the closed-loop as the open-loop responses time.

The PRBS signal has low frequency power spectral amplitude given by

$$\Phi_{uu}(\omega) = \frac{\alpha^2(T+1)}{T}, \quad (3.52)$$

and is reduced to half of this value at  $\omega = 2.8/T_c$ .



The lowest frequency perturbed is given by  $k = 1$  in (3.50), which is

$$\omega_{low} = \frac{2\pi}{TT_c}. \quad (3.53)$$

So that, for practical purpose, the PRBS perturbs the system over

$$\frac{2\pi}{TT_c} < \omega_{PRBS} < \frac{2.8}{T_c}. \quad (3.54)$$

Although there would be some perturbation outside this range, but it decays rapidly with frequency.

From (3.51) and (3.54), it can be concluded that

$$T_c = \frac{2.8\tau}{\alpha}, \text{ and } T = \frac{2\pi\alpha\beta}{2.8} \approx 2\alpha\beta. \quad (3.55)$$

Thus the clock tick time is determined by the highest frequency of excitation desired, and the period of the PRBS is determined by the lowest frequency.

The number of shift registers used to generate the PRBS signal,  $n$ , commonly is chosen so that it is closest to  $T = 2^n - 1$ .

Since the signal will be sampled, the sampling rate also has to be considered. According to the sampling theorem, the signal should be sampled at sampling rate that is

$$T_s < \frac{\pi}{\omega_{high}}, \quad (3.56)$$

where  $\omega_{high}$  (rad/sec) is the highest frequency present in the signal. Then from (3.54) and (3.56), the following is implied

$$T_s < \frac{\pi}{2.8} T_c. \quad (3.57)$$

In practice, it is recommended to use a sampling time at least 10 times smaller than the clock tick time [92].

The amplitude of PRBS is selected based on the range over which the controlled variable is allowed to change using the knowledge about the process gain. Normally, the higher

amplitude will produce the higher signal-to-noise ratio (*S/N ratio*). The signal to noise ratio is estimated from

$$S / N \text{ ratio} = \frac{\sigma_s^2}{\sigma_N^2}, \quad (3.58)$$

where  $\sigma_s$  is the standard deviation of signal, and  $\sigma_N$  is the standard deviation of noise. A signal to noise ratio of at least 6 is recommended by practitioners [92].

The information about PRBS signals will be used to design the input signals for system identification of the plant. The input signals, which will be transmitted to the three control valves (PCV202, FCV211, and PCV212), are designed independently. The design is based on the step responses data and the FOPDT transfer functions that have been obtained during preliminary test. Considering that there are two outputs of plant, which each of them has different plant dynamics for the respective input, then the plant response that has the longest time constant will be considered for designing the PRBS signal. Table 3.4 shows the parameters used in the design of the PRBS signals in this regard.

**Table 3.4:** PRBS signals design of the three plant inputs

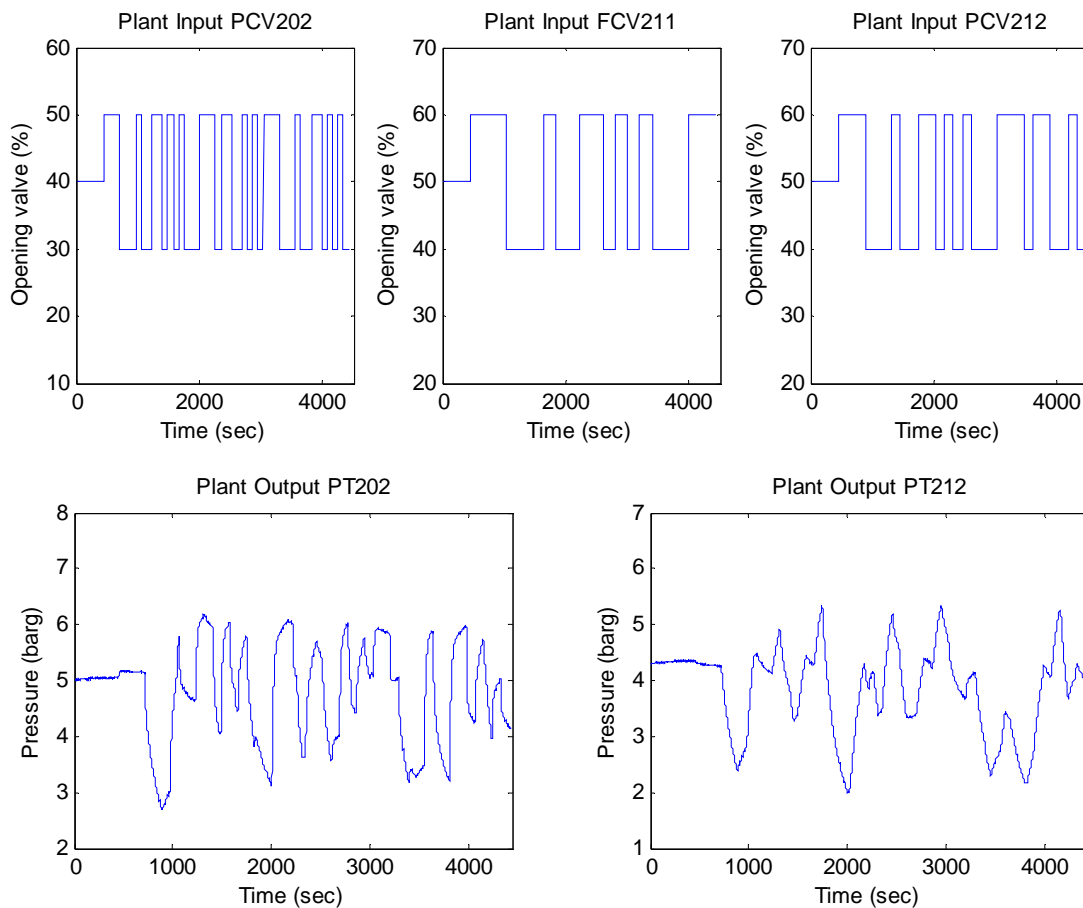
PRBS parameter	Plant input		
	PCV202	FCV211	PCV212
$\omega_{\max}$	0.0328 rad/sec	0.0143 rad/sec	0.0196 rad/sec
$\omega_{\min}$	0.0055 rad/sec	0.0024 rad/sec	0.0033 rad/sec
$T_c$	86 sec	196 sec	143 sec
$n$	4	4	4
T	15	15	15

During a preliminary test, a 20% of input signal amplitude was applied. This setting results a large signal-to-noise ratio. Examining the gain of the plant and noise statistics, then a 10% of input signal amplitude,  $a$ , is considered to obtain an adequate signal to noise ratio. Since the lowest clock tick time is 86 seconds, the recommended time sampling is at least 8.6 seconds. In this work, the sampling time chosen is one second.

### 3.5.2.3 Identification process

During testing, the three different PRBS signals are transmitted to the respective control valves simultaneously, and the pressure changes in the upstream and downstream are recorded as plant outputs. This input-output data is shown in Figure 3.15. The total recorded data is 4500, which is then divided into two parts: one part for identification, and the other part for validation purpose.

Similar with the development of Model 1, the model structure of the identified system in this particular case remains using the structure in (3.38). Hence, the input and output vectors of the resulted model are equal to the input and output vectors of Model 1.



**Figure 3.15:** PRBS signals and plant responses for the development of Model 2.

The suggested order by the N4SID algorithm, in this particular case, is  $n = 4$ . From the Hankel singular values inspection, it is illustrated that the first four order states store the most of “energy” in the system. The relative estimation indexes also indicate that the

minimum simulation errors are also given by the order  $n = 4$ . Hence, the state-space model of gaseous pilot plant for Model 2 is then chosen to be of 4<sup>th</sup>-order.

#### 3.5.2.4 Identification result and model validation

To obtain the parameters of Model 2 the input-output data shown in Figure 3.15 are yet divided into two sets of data. The first data set would be used as the identification data of N4SID algorithm to determine the element of coefficient matrices of Model 2. Considering the signal-to-noise ratio ( $S/N$  ratio) of the identification data, and since there is no outlier detected in the observation, the pre-treatment of the input-output data such as noise filtering and outlier removal are not conducted.

From MIMO identification process using N4SID algorithm, the coefficient matrices of Model 2 are obtained. The model is in discrete-time form with the sampling time equals to one second. These coefficient matrices are shown in (3.59) until (3.62):

$$\mathbf{A} = \begin{bmatrix} 0.9940 & 0.0012 & -0.0071 & -0.0021 \\ 0.0217 & 0.9716 & 0.0398 & -0.0288 \\ -0.0126 & 0.0031 & 0.9702 & -0.0114 \\ 0.0222 & -0.0129 & 0.0404 & 0.9589 \end{bmatrix}; \quad (3.59)$$

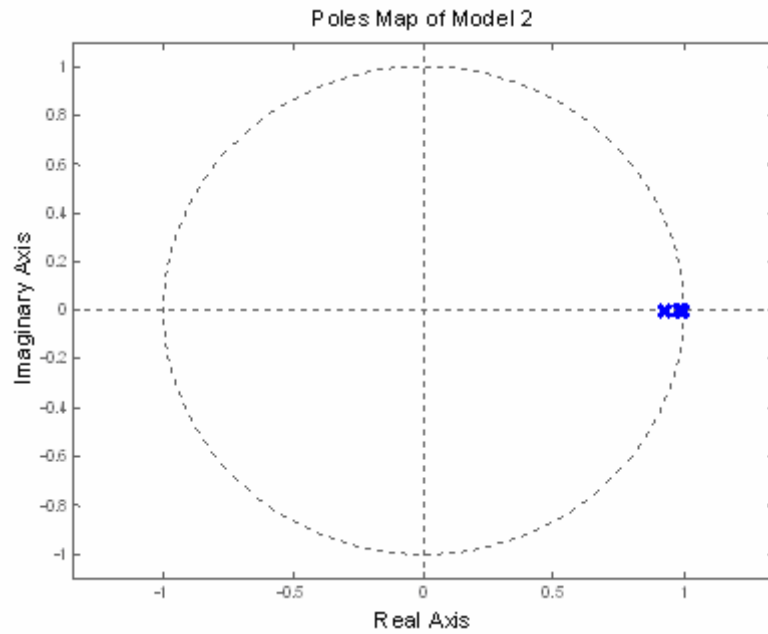
$$\mathbf{B} = \begin{bmatrix} 0.000038 & -0.000008 & -0.000025 \\ -0.000244 & 0.000134 & 0.000041 \\ 0.000100 & -0.000050 & -0.000054 \\ -0.000222 & 0.000100 & 0.000065 \end{bmatrix}; \quad (3.60)$$

$$\mathbf{C} = \begin{bmatrix} 37.126 & -13.183 & -0.279 & 0.147 \\ 38.803 & 5.462 & 0.114 & 0.029 \end{bmatrix}; \quad (3.61)$$

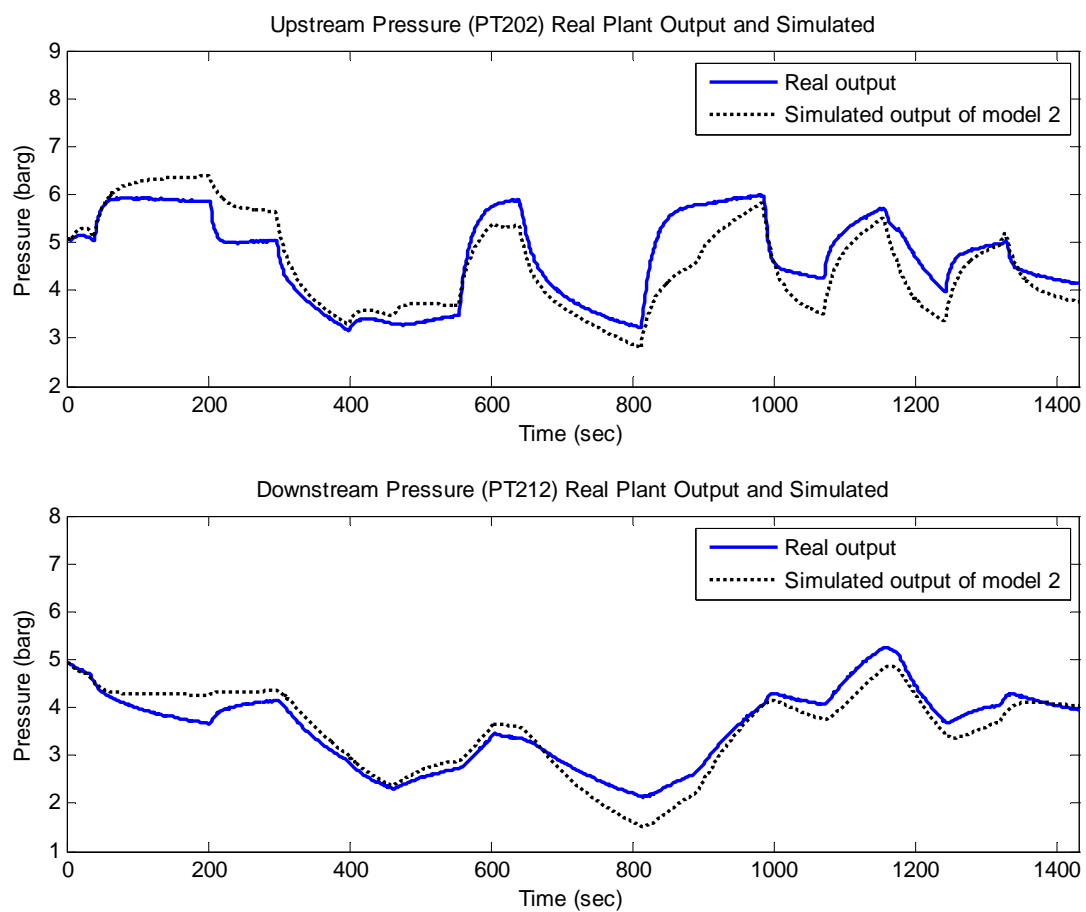
$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.62)$$

The poles of this particular plant model (the eigenvalues of  $\mathbf{A}$  matrix of Model 2), which are shown in Figure 3.16, are inside the unit circle denote that the system is stable.

Figure 3.17 shows the comparisons of the real outputs of plant (solid line) and the simulation outputs which are generated by Model 2 (dotted line) for the validation data. The simulation starts initially with an estimated state. The best fit criterion of Model 2 for identification and validation data are shown in Table 3.5.



**Figure 3.16:** Location of the poles of Model 2.



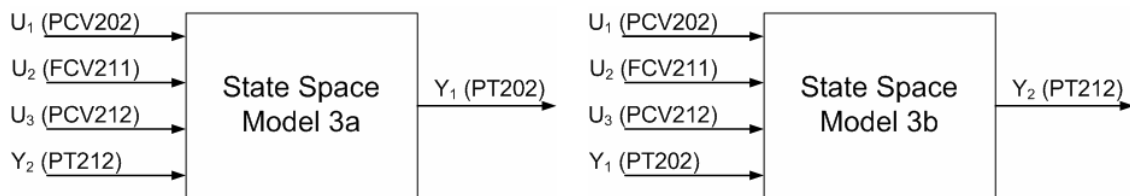
**Figure 3.17:** The comparison of real output and simulated output generated by Model 2 for upstream pressure (top) and downstream pressure (bottom).

**Table 3.5:** The performance of Model 2.

Plant output	Best fit criterion (%)	
	Identification data	Validation data
PT202 (upstream pressure)	73.52	46.25
PT212 (downstream pressure)	67.13	60.49

### 3.5.3 Model 3

As briefly mentioned before, the third approach for developing the empirical model of the gaseous pilot plant is by using (3.39) and (3.40) as the model structures of the identified system. Since a variable cannot be the input and output of identification at the same time, then the empirical modeling of such a plant, in this particular case, is considered as constructing two MISO models, which each model has four inputs. The input variables of the first MISO model are: PCV202, FCV211, PCV212, and PT212 respectively, while its output is PT202. For the second MISO model, the input variables are: PCV202, FCV211, PCV212, and PT202 respectively, while the output variable is PT212. The structures of these two MISO models with their respective variables are illustrated in Figure 3.18.



**Figure 3.18:** The schematic diagram of the second model structure which consists of two MISO models.

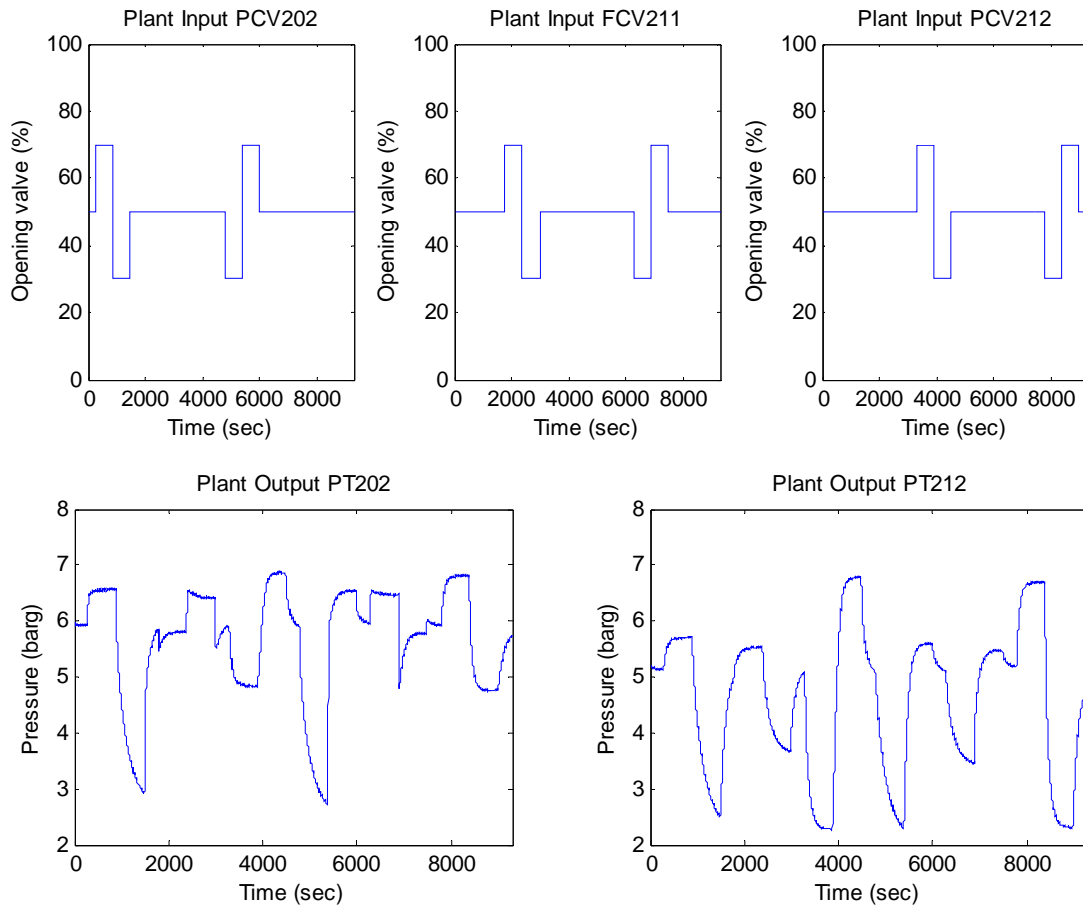
#### 3.5.3.1 MISO state-space system identification

For identification purpose, a similar procedure to the plant testing in sub-section 3.5.1.1 using step input signals is conducted. The period of step input signals are prolonged, since it is expected that the interactions between the upstream and the downstream pressures can be well reflected in the input-output data. The total of recorded data is 9300 with one second sampling time. The length of each step signal is 600 seconds, or three times the

length of step signals which are used in the development of Model 1. The input-output data used in the development of Model 3 are illustrated in Figure 3.19.

The total recorded data is then divided into two: the first half is used for identification purpose, and the rest is used for model validation.

The identification processes of the two MISO models are performed independently. From the singular value decompositions, the N4SID algorithm suggests that the appropriate orders of the both MISO models are  $n = 2$ . From the Hankel singular values inspections, the most of “energy” of the both models are stored in their first two states. The relative estimation indexes also indicate that the minimum simulation errors of the both systems are obtained when each of the models is having the order  $n = 2$ . Hence, both of the MISO state-space models are then chosen to be of 2<sup>nd</sup>-order.



**Figure 3.19:** Input signals and plant responses for the development of Model 3.

### 3.5.3.2 Combination of MISO models

The two MISO models that have been obtained are necessary to be reconstructed into a single MIMO model, so that it can be used in the implementation of a multivariable control system. For that purpose, the following procedure is proposed in this work.

Let the two obtained MISO models are named Model 3a and Model 3b. Model 3a, which is used to simulate the upstream pressure, has structure as follow:

$$\begin{aligned}\mathbf{x}_1(k+1) &= \mathbf{A}_1\mathbf{x}_1(k) + \mathbf{B}_1\mathbf{u}_1(k) \\ \mathbf{y}_1(k) &= \mathbf{C}_1\mathbf{x}_1(k) + \mathbf{D}_1\mathbf{u}_1(k).\end{aligned}\quad (3.63)$$

From the order estimation, both of Model 3a and Model 3b are selected as 2<sup>nd</sup>-order systems. Hence, Model 3a can be written in detail as

$$\begin{bmatrix} x_{1-1}(k+1) \\ x_{1-2}(k+1) \end{bmatrix} = \begin{bmatrix} a_{1-11} & a_{1-12} \\ a_{1-21} & a_{1-22} \end{bmatrix} \begin{bmatrix} x_{1-1}(k) \\ x_{1-2}(k) \end{bmatrix} + \begin{bmatrix} b_{1-11} & b_{1-12} & b_{1-13} & b_{1-14} \\ b_{1-21} & b_{1-22} & b_{1-23} & b_{1-24} \end{bmatrix} \begin{bmatrix} \text{PCV202}(k) \\ \text{FCV211}(k) \\ \text{PCV212}(k) \\ \text{PT212}(k) \end{bmatrix}; \quad (3.64)$$

$$y_1(k) = \begin{bmatrix} c_{1-11} & c_{1-12} \end{bmatrix} \begin{bmatrix} x_{1-1}(k) \\ x_{1-2}(k) \end{bmatrix} + \begin{bmatrix} d_{1-11} & d_{1-12} & d_{1-13} & d_{1-14} \end{bmatrix} \begin{bmatrix} \text{PCV202}(k) \\ \text{FCV211}(k) \\ \text{PCV212}(k) \\ \text{PT212}(k) \end{bmatrix}, \quad (3.65)$$

with  $a_{1-ij}$ ,  $b_{1-ij}$ ,  $c_{1-ij}$ , and  $d_{1-ij}$  represent the element of  $i^{\text{th}}$ -row and  $j^{\text{th}}$ -column of the respective coefficient matrices, and  $x_{1-i}$  represents the  $i^{\text{th}}$ -element of the state vectors, which are obtained from the subspace system identification algorithm.  $y_1$  represents the output of the Model 3a, which is the upstream pressure (PT202).

In the same way, Model 3b can be written as

$$\begin{aligned}\mathbf{x}_2(k+1) &= \mathbf{A}_2\mathbf{x}_2(k) + \mathbf{B}_2\mathbf{u}_2(k) \\ \mathbf{y}_2(k) &= \mathbf{C}_2\mathbf{x}_2(k) + \mathbf{D}_2\mathbf{u}_2(k),\end{aligned}\quad (3.66)$$



or in detail as

$$\begin{bmatrix} x_{2-1}(k+1) \\ x_{2-2}(k+1) \end{bmatrix} = \begin{bmatrix} a_{2-11} & a_{2-12} \\ a_{2-21} & a_{2-22} \end{bmatrix} \begin{bmatrix} x_{2-1}(k) \\ x_{2-2}(k) \end{bmatrix} + \begin{bmatrix} b_{2-11} & b_{2-12} & b_{2-13} & b_{2-14} \\ b_{2-21} & b_{2-22} & b_{2-23} & b_{2-24} \end{bmatrix} \begin{bmatrix} \text{PCV202}(k) \\ \text{FCV211}(k) \\ \text{PCV212}(k) \\ \text{PT202}(k) \end{bmatrix}; \quad (3.67)$$

$$y_2(k) = \begin{bmatrix} c_{2-11} & c_{2-12} \end{bmatrix} \begin{bmatrix} x_{2-1}(k) \\ x_{2-2}(k) \end{bmatrix} + \begin{bmatrix} d_{2-11} & d_{2-12} & d_{2-13} & d_{2-14} \end{bmatrix} \begin{bmatrix} \text{PCV202}(k) \\ \text{FCV211}(k) \\ \text{PCV212}(k) \\ \text{PT202}(k) \end{bmatrix} \quad (3.68)$$

with  $a_{2-ij}$ ,  $b_{2-ij}$ ,  $c_{2-ij}$ , and  $d_{2-ij}$  represent the element of  $i^{\text{th}}$ -row and  $j^{\text{th}}$ -column of the respective coefficient matrices, and  $x_{2-i}$  represents the  $i^{\text{th}}$ -element of the state vectors, which are obtained from the subspace system identification algorithm.  $y_2$  represents the output of the Model 3b, which is the downstream pressure (PT212).

The method to combine Model 3a and Model 3b into a single MIMO model can be explained in these three following steps:

Step 1: Constructing the new input, state, and output vectors.

Model 3a has four inputs, where three of them are also used as the inputs for Model 3b, i.e., the inlet control valve (PCV202), the middle control valve (FCV211), and the outlet control valve (PCV212). Since these inputs are used by both of MISO models, they are called the common inputs. In the matrix notation, the common inputs of both models can be written as

$$\mathbf{u}_c = [\text{PCV202} \quad \text{FCV211} \quad \text{PCV212}]^T. \quad (3.69)$$

The input which is used only by one model is called the individual input. For Model 3a, its individual input is

$$\mathbf{u}_{ind-1} = [\text{PT212}], \quad (3.70)$$

and for Model 3b, its individual input is

$$\mathbf{u}_{ind-2} = [\text{PT202}]. \quad (3.71)$$

The new input vector is defined as

$$\begin{aligned}\mathbf{u} &= \begin{bmatrix} \mathbf{u}_c^T & \mathbf{u}_{ind-1} & \mathbf{u}_{ind-2} \end{bmatrix}^T \\ &= [\text{PCV202} \quad \text{FCV211} \quad \text{PCV212} \quad \text{PT212} \quad \text{PT202}]^T.\end{aligned}\quad (3.72)$$

The new state vector is obtained by combining the state vector of Model 3a,  $\mathbf{x}_1$ , and the state vector of Model 3b,  $\mathbf{x}_2$ , as follow:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{bmatrix}^T = \begin{bmatrix} x_{1-1} & x_{1-2} & x_{2-1} & x_{2-2} \end{bmatrix}^T. \quad (3.73)$$

And the new output vector consist of the output of both models, which are arranged in the matrix notation as

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T. \quad (3.74)$$

Step 2: Splitting the input and feedthrough coefficients matrices (**B** and **D**).

In the state-space model, there are two coefficient matrices which coupled with the input vector. These two coefficient matrices are the **B** matrix, also called the input coefficient matrix, and the **D** matrix, also called the feedthrough coefficient matrix.

Like the input vector, each of the input and feedthrough coefficient matrices is also necessary to be divided into two new matrices. The first new matrix contains the elements which are coupled with the common inputs,  $\mathbf{u}_c$ , and the second one contains the elements which are coupled with the individual input,  $\mathbf{u}_{ind}$ .

For Model 3a in (3.64), the common inputs are located in the first three rows and the individual input is located in the last row. So that the input and feedthrough coefficient matrices of Model 3a,  $\mathbf{B}_1$  and  $\mathbf{D}_1$ , can be simply divided by taking the first three columns as the coefficient matrices which are coupled with the common inputs,  $\mathbf{B}_{11}$  and  $\mathbf{D}_{11}$ , and the last column as the coefficient matrices which are coupled with the individual input,  $\mathbf{B}_{12}$  and  $\mathbf{D}_{12}$ . In detail, it can be expressed as

$$\mathbf{B}_1 = [\mathbf{B}_{11} \quad \mathbf{B}_{12}]; \quad \mathbf{B}_{11} = \begin{bmatrix} b_{1-11} & b_{1-12} & b_{1-13} \\ b_{1-21} & b_{1-22} & b_{1-23} \end{bmatrix}; \quad \mathbf{B}_{12} = \begin{bmatrix} b_{1-14} \\ b_{1-24} \end{bmatrix}, \quad (3.75)$$

and

$$\mathbf{D}_1 = [\mathbf{D}_{11} \quad \mathbf{D}_{12}]; \quad \mathbf{D}_{11} = [d_{1-11} \quad d_{1-12} \quad d_{1-13}]; \quad \mathbf{D}_{12} = [d_{1-14}]. \quad (3.76)$$

Inline with Model 3a, since the common inputs of Model 3b are also located in the first three rows and the individual input is located in the last row in (3.67), the input and feedthrough coefficient matrices of Model 3b,  $\mathbf{B}_2$  and  $\mathbf{D}_2$ , can be simply split as

$$\mathbf{B}_2 = [\mathbf{B}_{21} \quad \mathbf{B}_{22}]; \quad \mathbf{B}_{21} = \begin{bmatrix} b_{2-11} & b_{2-12} & b_{2-13} \\ b_{2-21} & b_{2-22} & b_{2-23} \end{bmatrix}; \quad \mathbf{B}_{22} = \begin{bmatrix} b_{2-14} \\ b_{2-24} \end{bmatrix}, \quad (3.77)$$

and

$$\mathbf{D}_2 = [\mathbf{D}_{21} \quad \mathbf{D}_{22}]; \quad \mathbf{D}_{21} = [d_{2-11} \quad d_{2-12} \quad d_{2-13}]; \quad \mathbf{D}_{22} = [d_{2-14}]. \quad (3.78)$$

Step 3: Constructing the new coefficient matrices.

The new state coefficient matrix,  $\mathbf{A}$ , is obtained by locating the state coefficient matrices of Model 3a and Model 3b in the pseudo-diagonal formation as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} a_{1-11} & a_{1-12} & 0 & 0 \\ a_{1-21} & a_{1-22} & 0 & 0 \\ 0 & 0 & a_{2-11} & a_{2-12} \\ 0 & 0 & a_{2-21} & a_{2-22} \end{bmatrix}. \quad (3.79)$$

In the same way, the new output coefficient matrix,  $\mathbf{C}$ , is obtained:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 \end{bmatrix} = \begin{bmatrix} c_{1-11} & c_{1-12} & 0 & 0 \\ 0 & 0 & c_{2-11} & c_{2-12} \end{bmatrix}. \quad (3.80)$$

Since the input and feedthrough coefficient matrices consist of the coefficient of the common inputs and individual inputs, then the new input and feedthrough coefficient matrices,  $\mathbf{B}$  and  $\mathbf{D}$ , are obtained in a different way. Since the common inputs in (3.72) are located in the first three columns of  $\mathbf{u}$  vector and the rest are occupied by individual inputs, then the elements which are coupled with the common inputs have to be located in the first three columns of  $\mathbf{B}$  and  $\mathbf{D}$  matrices, and the rest are occupied with the elements which are coupled with the individual inputs, arranged in pseudo-diagonal formation, as follow:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{0} \\ \mathbf{B}_{21} & \mathbf{0} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} b_{1-11} & b_{1-12} & b_{1-13} & b_{1-14} & 0 \\ b_{1-21} & b_{1-22} & b_{1-23} & b_{1-24} & 0 \\ b_{2-11} & b_{2-12} & b_{2-13} & 0 & b_{2-14} \\ b_{2-21} & b_{2-22} & b_{2-23} & 0 & b_{2-24} \end{bmatrix}, \quad (3.81)$$

and

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{0} \\ \mathbf{D}_{21} & \mathbf{0} & \mathbf{D}_{22} \end{bmatrix} = \begin{bmatrix} d_{1-11} & d_{1-12} & d_{1-13} & d_{1-14} & 0 \\ d_{2-11} & d_{2-12} & d_{2-13} & 0 & d_{2-24} \end{bmatrix}. \quad (3.82)$$

From three steps mentioned above, the all elements of the MIMO state-space model are obtained. So that, the final model of Model 3 can now be constructed in a MIMO state-space notation as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k). \end{aligned} \quad (3.83)$$

In the complete form, Model 3 can be written as

$$\begin{aligned} \begin{bmatrix} x_{1-1}(k+1) \\ x_{1-2}(k+1) \\ x_{2-1}(k+1) \\ x_{2-2}(k+1) \end{bmatrix} &= \begin{bmatrix} a_{1-11} & a_{1-12} & 0 & 0 \\ a_{1-21} & a_{1-22} & 0 & 0 \\ 0 & 0 & a_{2-11} & a_{2-12} \\ 0 & 0 & a_{2-21} & a_{2-22} \end{bmatrix} \begin{bmatrix} x_{1-1}(k) \\ x_{1-2}(k) \\ x_{2-1}(k) \\ x_{2-2}(k) \end{bmatrix} \\ &+ \begin{bmatrix} b_{1-11} & b_{1-12} & b_{1-13} & b_{1-14} & 0 \\ b_{1-21} & b_{1-22} & b_{1-23} & b_{1-24} & 0 \\ b_{2-11} & b_{2-12} & b_{2-13} & 0 & b_{2-14} \\ b_{2-21} & b_{2-22} & b_{2-23} & 0 & b_{2-24} \end{bmatrix} \begin{bmatrix} \text{PCV202}(k) \\ \text{FCV211}(k) \\ \text{PCV212}(k) \\ \text{PT212}(k) \\ \text{PT202}(k) \end{bmatrix}; \end{aligned} \quad (3.84)$$

$$\begin{aligned} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} &= \begin{bmatrix} c_{1-11} & c_{1-12} & 0 & 0 \\ 0 & 0 & c_{2-11} & c_{2-12} \end{bmatrix} \begin{bmatrix} x_{1-1}(k) \\ x_{1-2}(k) \\ x_{2-1}(k) \\ x_{2-2}(k) \end{bmatrix} \\ &+ \begin{bmatrix} d_{1-11} & d_{1-12} & d_{1-13} & d_{1-14} & 0 \\ d_{2-11} & d_{2-12} & d_{2-13} & 0 & d_{2-14} \end{bmatrix} \begin{bmatrix} \text{PCV202}(k) \\ \text{FCV211}(k) \\ \text{PCV212}(k) \\ \text{PT212}(k) \\ \text{PT202}(k) \end{bmatrix}. \end{aligned} \quad (3.85)$$

The state-space model in (3.84) and (3.85) can be expanded in the six following equations:

$$y_1(k) = c_{1-11}x_{1-1}(k) + c_{1-12}x_{1-2}(k) + d_{1-11}\text{PCV202}(k) + d_{1-12}\text{FCV211}(k) + d_{1-13}\text{PCV212}(k) + d_{1-14}\text{PT212}(k), \quad (3.86)$$

$$x_{1-1}(k+1) = a_{1-11}x_{1-1}(k) + a_{1-12}x_{1-2}(k) + b_{1-11}\text{PCV202}(k) + b_{1-12}\text{FCV211}(k) + b_{1-13}\text{PCV212}(k) + b_{1-14}\text{PT212}(k), \quad (3.87)$$

$$x_{1-2}(k+1) = a_{1-21}x_{1-1}(k) + a_{1-22}x_{1-2}(k) + b_{1-21}\text{PCV202}(k) + b_{1-22}\text{FCV211}(k) + b_{1-23}\text{PCV212}(k) + b_{1-24}\text{PT212}(k), \quad (3.88)$$

and

$$y_2(k) = c_{2-11}x_{2-1}(k) + c_{2-12}x_{2-2}(k) + d_{2-11}\text{PCV202}(k) + d_{2-12}\text{FCV211}(k) + d_{2-13}\text{PCV212}(k) + d_{2-14}\text{PT202}(k), \quad (3.89)$$

$$x_{2-1}(k+1) = a_{2-11}x_{2-1}(k) + a_{2-12}x_{2-2}(k) + b_{2-11}\text{PCV202}(k) + b_{2-12}\text{FCV211}(k) + b_{2-13}\text{PCV212}(k) + b_{2-14}\text{PT202}(k), \quad (3.90)$$

$$x_{2-2}(k+1) = a_{2-21}x_{2-1}(k) + a_{2-22}x_{2-2}(k) + b_{2-21}\text{PCV202}(k) + b_{2-22}\text{FCV211}(k) + b_{2-23}\text{PCV212}(k) + b_{2-24}\text{PT202}(k). \quad (3.91)$$

It may be noted that the equations (3.86) to (3.91) returns the value of MISO models in (3.64) and (3.67) which are given by the N4SID subspace system identification algorithm. It proves that even though Model 3a and Model 3b may be estimated in the different state bases, this technique remains relevant.

Additionally, it is also possible to combine more than two MISO models into a single MIMO model using this proposed technique. Suppose that  $l$  MISO state-space models are developed, which each model has the input vector:  $\mathbf{u}_m = [\mathbf{u}_c^T \quad \mathbf{u}_{ind-m}^T]^T$ , with  $m$  is the index of the MISO model ( $m = 1, 2, \dots, l$ ), consist of  $c$  common inputs which are used in all MISO model:  $\mathbf{u}_c = [u_{c-1} \quad u_{c-2} \quad \dots \quad u_{c-c}]^T$ , and  $h$  individual inputs,  $\mathbf{u}_{ind-m} = [u_{ind-m-1} \quad u_{ind-m-2} \quad \dots \quad u_{ind-m-h}]^T$ , which are only used in the  $m^{\text{th}}$ -model. Each MISO model may have different number of  $h$ . Let the  $m^{\text{th}}$ -model has state coefficient matrices with their respective dimensions as follows:

$$\mathbf{A}_m : n_x \times n_x; \mathbf{B}_m : n_x \times (c+h); \mathbf{C}_m : 1 \times n_x; \mathbf{D}_m : 1 \times (c+h), \quad (3.92)$$

with  $n_x$  is the size of state vector,  $c$  is the number of the common inputs, and  $h$  is the number of the individual inputs in the respective MISO model. The different MISO models may also have the different values of  $n_x$  and  $h$ .

In order to combine  $l$  MISO models into a MIMO model, the new input, state, and output vectors are necessarily to be constructed as follows:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_c^T & \mathbf{u}_{ind-1}^T & \mathbf{u}_{ind-2}^T & \cdots & \mathbf{u}_{ind-l}^T \end{bmatrix}^T; \quad (3.93)$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_l^T \end{bmatrix}^T; \quad (3.94)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_l \end{bmatrix}^T. \quad (3.95)$$

Then, the input and the feedthrough coefficient matrices of each MISO model,  $\mathbf{B}_m$  and  $\mathbf{D}_m$ , should be split become two matrices with the same principle. The first matrix is for the elements which are coupled with the common inputs while the second matrix is for the elements which are coupled to the individual input, which can be written as

$$\mathbf{B}_m = [\mathbf{B}_{m1} \quad \mathbf{B}_{m2}]; \quad \mathbf{D}_m = [\mathbf{D}_{m1} \quad \mathbf{D}_{m2}], \quad (3.96)$$

with

$$\mathbf{B}_{m1} = \begin{bmatrix} \mathbf{B}_m(1,1) & \cdots & \mathbf{B}_m(1,c) \\ \vdots & & \vdots \\ \mathbf{B}_m(n_x,1) & \cdots & \mathbf{B}_m(n_x,c) \end{bmatrix}; \quad \mathbf{B}_{m2} = \begin{bmatrix} \mathbf{B}_m(1,c+1) & \cdots & \mathbf{B}_m(1,c+h) \\ \vdots & & \vdots \\ \mathbf{B}_m(n_x,c+1) & \cdots & \mathbf{B}_m(n_x,c+h) \end{bmatrix}; \quad (3.97)$$

and

$$\mathbf{D}_{m1} = [\mathbf{D}_m(1,1) \quad \cdots \quad \mathbf{D}_m(1,c)]; \quad \mathbf{D}_{m2} = [\mathbf{D}_m(1,c+1) \quad \cdots \quad \mathbf{D}_m(1,c+h)]. \quad (3.98)$$

Then the new state coefficient matrices of ( $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$ ) are necessary to be formed as shown in (3.99) until (3.102):

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_m \end{bmatrix}; \quad (3.99)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B}_{21} & \mathbf{0} & \mathbf{B}_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{B}_{m1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{m2} \end{bmatrix}; \quad (3.100)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_m \end{bmatrix}; \quad (3.101)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{D}_{21} & \mathbf{0} & \mathbf{D}_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{D}_{m1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{D}_{m2} \end{bmatrix}. \quad (3.102)$$

So, it is also possible to use this proposed technique to develop a MIMO model of the interacting series processes which are constructed from more than two MISO models.

### 3.5.3.3 Identification result and model validation

From the identification processes using N4SID algorithm, two MISO models named Model 3a and Model 3b are obtained. The models are presented in the discrete-time domain with one second of sampling time, and having the linear time-invariant variables.

Model 3a has the coefficient matrices as shown in (3.103) to (3.106):

$$\mathbf{A}_1 = \begin{bmatrix} 0.9953 & -0.0287 \\ -0.0123 & 0.8978 \end{bmatrix}; \quad (3.103)$$

$$\mathbf{B}_1 = \begin{bmatrix} 0.000179 & -0.000228 & 0.000107 & 0.003940 \\ 0.000644 & -0.000802 & 0.000358 & 0.013803 \end{bmatrix}; \quad (3.104)$$

$$\mathbf{C}_1 = [21.64 \quad -0.10]; \quad (3.105)$$

$$\mathbf{D}_1 = [0 \quad 0 \quad 0 \quad 0], \quad (3.106)$$

with the input and the output vectors are:

$$\mathbf{u}_1 = [\text{PCV202} \quad \text{FCV211} \quad \text{PCV212} \quad \text{PT212}]^T; \quad (3.107)$$

$$\mathbf{y}_1 = [\text{PT202}]. \quad (3.108)$$

The coefficient matrices for Model 3b are given in (3.109) to (3.112):

$$\mathbf{A}_2 = \begin{bmatrix} 0.9892 & -0.0097 \\ -0.0028 & 0.9897 \end{bmatrix}; \quad (3.109)$$

$$\mathbf{B}_2 = \begin{bmatrix} 0.000012 & 0.000032 & -0.000045 & 0.000339 \\ 0.000006 & 0.000015 & -0.000026 & 0.000134 \end{bmatrix}; \quad (3.110)$$

$$\mathbf{C}_2 = [28.07 \quad -0.10]; \quad (3.111)$$

$$\mathbf{D}_2 = [0 \quad 0 \quad 0 \quad 0], \quad (3.112)$$

with the input and the output vectors for Model 3b are:

$$\mathbf{u}_2 = [\text{PCV202} \quad \text{FCV211} \quad \text{PCV212} \quad \text{PT202}]^T; \quad (3.113)$$

$$\mathbf{y}_2 = [\text{PT212}]. \quad (3.114)$$

Using the technique explained in 3.5.3.2, Model 3a and Model 3b are then combined into a single MIMO state-space model, named as Model 3. The input vector of Model 3 is

$$\mathbf{u} = [\text{PCV202} \quad \text{FCV211} \quad \text{PCV212} \quad \text{PT212} \quad \text{PT202}]^T, \quad (3.115)$$

while its output vector is

$$\mathbf{y} = [\text{PT202} \quad \text{PT212}]^T. \quad (3.116)$$

Presented in discrete-time domain with one second of sampling time, the coefficient matrices of Model 3 are given in (3.117) to (3.120):

$$\mathbf{A} = \begin{bmatrix} 0.9953 & -0.0287 & 0 & 0 \\ -0.0123 & 0.8978 & 0 & 0 \\ 0 & 0 & 0.9892 & -0.0097 \\ 0 & 0 & -0.0028 & 0.9897 \end{bmatrix}; \quad (3.117)$$

$$\mathbf{B} = \begin{bmatrix} 0.000179 & -0.000228 & 0.000107 & 0.003940 & 0 \\ 0.000644 & -0.000802 & 0.000358 & 0.013803 & 0 \\ 0.000012 & 0.000032 & -0.000045 & 0 & 0.000339 \\ 0.000006 & 0.000015 & -0.000026 & 0 & 0.000134 \end{bmatrix}; \quad (3.118)$$

$$\mathbf{C} = \begin{bmatrix} 21.64 & -0.10 & 0 & 0 \\ 0 & 0 & 28.07 & -0.10 \end{bmatrix}; \quad (3.119)$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.120)$$

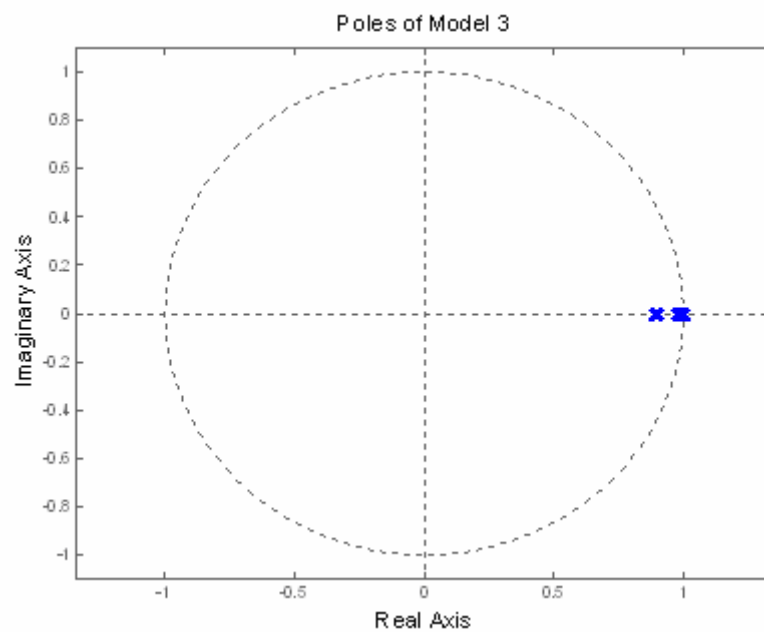


The poles of Model 3 (the eigenvalues of  $\mathbf{A}$  matrix), which are shown in Figure 3.20, are inside the unit circle denote that the system is stable. Yet, the poles close to 1, shown that the data set seems to contain a “co-integration” phenomenon [75].

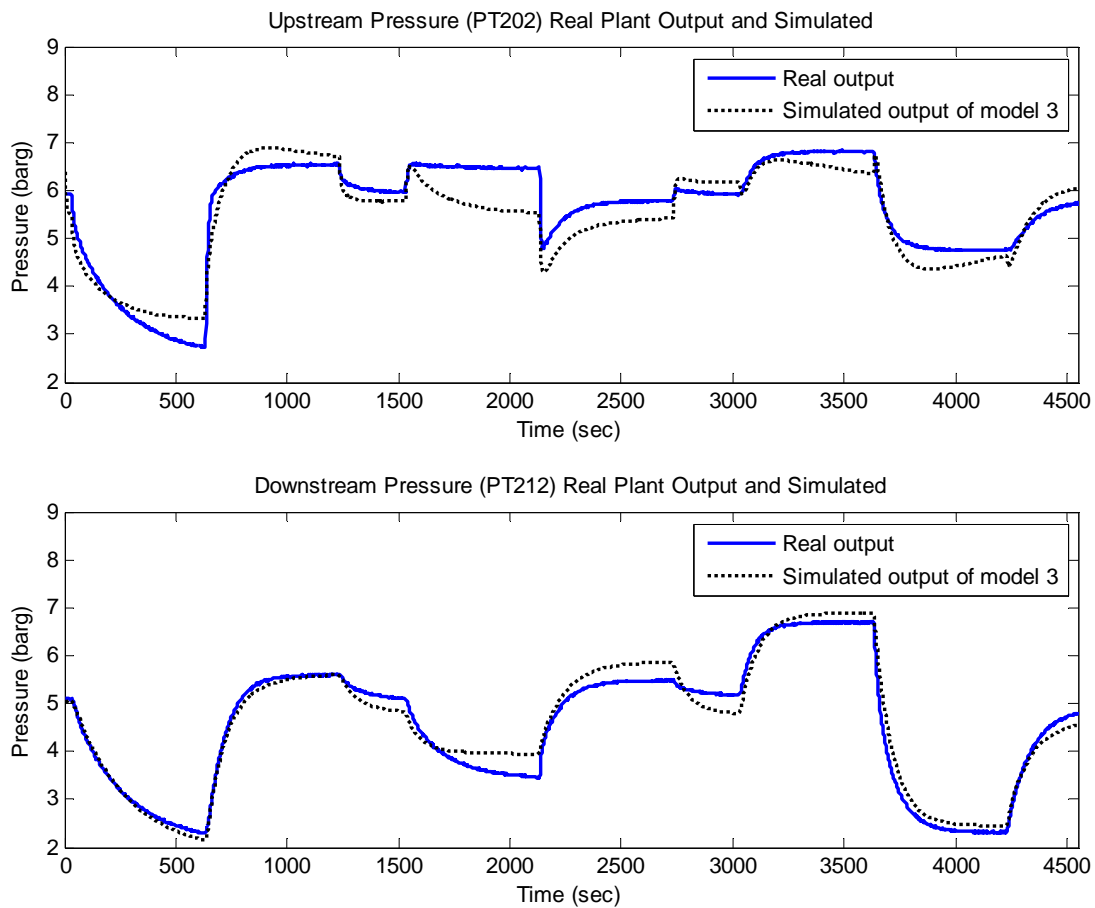
Figure 3.21 shows the comparisons of the real outputs of plant in the validation data and the simulated outputs which is generated by Model 3. The simulation starts initially with an estimated state. Similar to the previous developed models, the best fit criterion is used to measure the performance of Model 3. The best fit criterions of Model 3 for identification and validation data are shown in Table 3.6.

**Table 3.6:** The performance of Model 3

Plant output	Best fit criterion (%)	
	Identification data	Validation data
PT202 (upstream pressure)	75.12	57.91
PT212 (downstream pressure)	83.98	83.24



**Figure 3.20:** Location of the poles of Model 3.

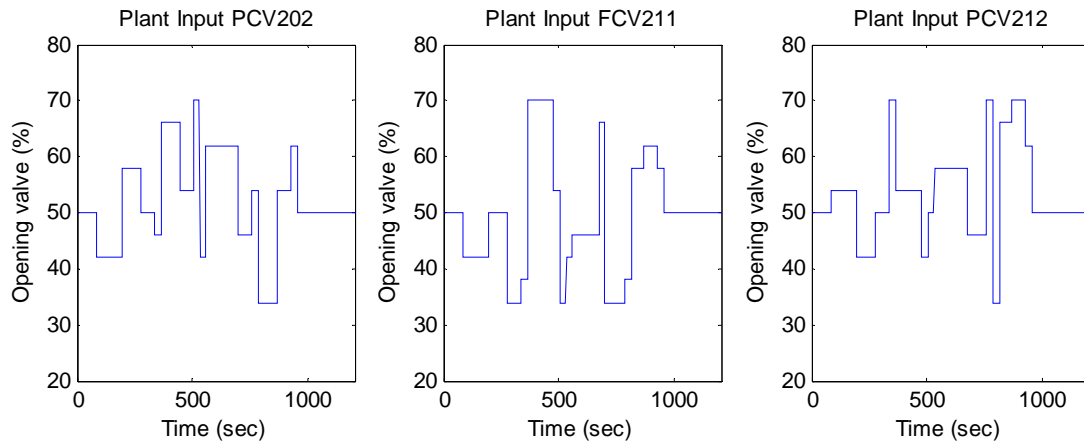


**Figure 3.21:** The comparison of real measured and simulated outputs generated by Model 3 for upstream pressure (top) and downstream pressure (bottom).

### 3.6 Analysis

Table 3.1, Table 3.5, and Table 3.6 show the best fit criterion of the constructed models. Model 3 has the highest best fit criterion in term of identification data. In term of validation data, the highest best fit criterion for downstream output data is also given by Model 3. The fitness of Model 1 for upstream identification data is quite poor. The entire model validations are conducted based on their own category: the models from step response are validated using step response data while the model from PRBS response is validated using PRBS response data. So far, the comparison is conducted only within the respective best fit criteria. To make a fair comparison of how good the models can reproduce the system behaviors and to examine the robustness against the plant nonlinearities, another kind of input signal which is not used earlier in the models development is worthwhile to be considered. In this work, the APRBS (Amplitude-modulation Pseudo-Random Binary Sequence) signal is selected. Beside the randomness

of its frequency, the amplitude of the APRBS signal is also modulated randomly. The three APRBS signals shown in Figure 3.22 are transmitted to the three control valves simultaneously as the plant inputs. The upstream and downstream responses are recorded with one second sampling time. The total of recorded data is 1200.



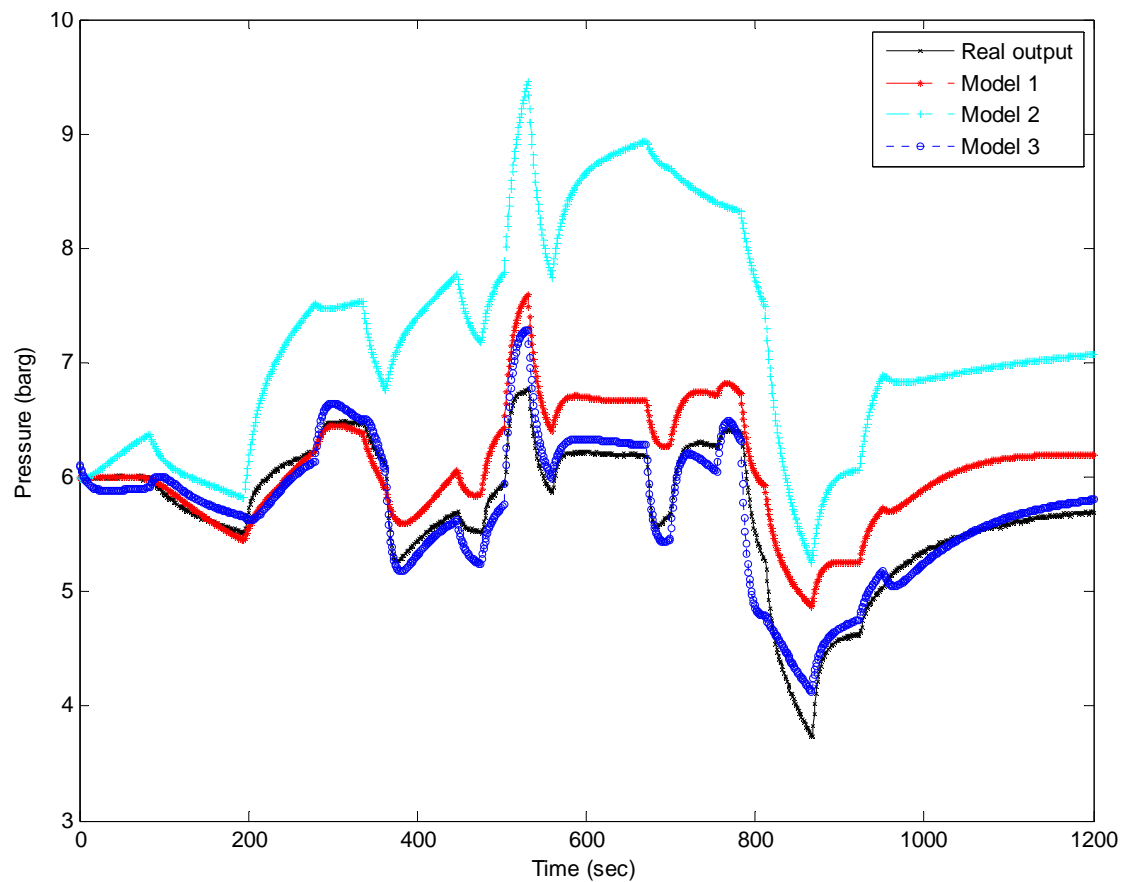
**Figure 3.22:** APRBS input of the three control valves for model validation.

The comparison of simulated outputs generated by the three models versus real outputs of plant from the APRBS response data for the upstream pressure is shown in Figure 3.23, while for the downstream pressure the comparison is shown in Figure 3.24. An estimated initial state of the respective model is assigned at the beginning of the simulation.

As can be observed in Figure 3.23, the best simulation outputs are given by Model 3. This model is able to reproduce the main dynamic characteristics of the plant at the given operating points and time horizons. For Model 1, the constant errors of prediction (offsets) start to occur after certain time horizons for both the upstream and downstream data. The simulated outputs of Model 1 return to the initial conditions when the input signals return to their initial conditions and remain constant. On the contrary, the validation data using APRBS input signals that were applied are not showing such linearities. Hence, it can be implied that Model 1 is not robust enough to cover the nonlinearities of the plant.

On another observation of the simulated outputs, the simulations of Model 3 exhibit the trend of the real measured outputs (validation data). By taking into account the measurement of the upstream output as one of the inputs to predict the downstream

output, and vice versa, the shiftiness of plant outputs, due to the nonlinearities, can be well approached by Model 3.

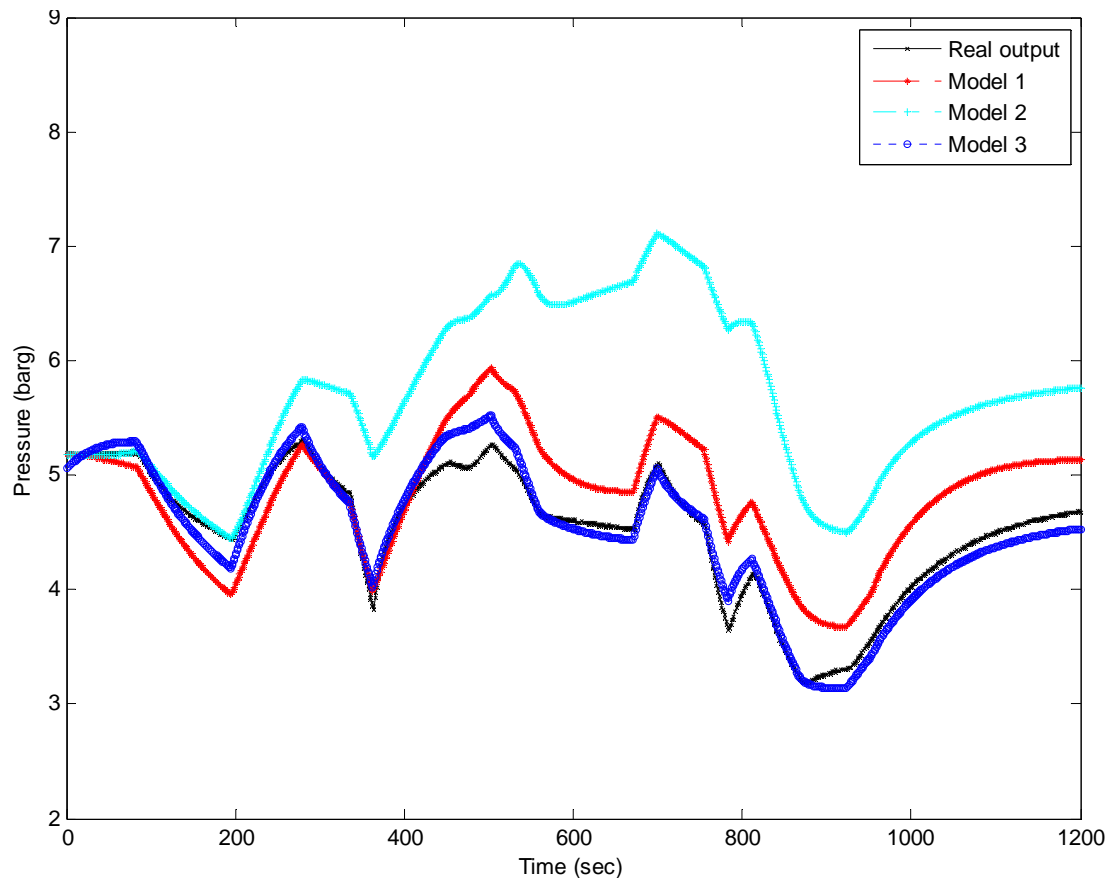


**Figure 3.23:** The comparison of the real output from APRBS response data and the simulated outputs generated by Model 1, Model 2, and Model 3 for the upstream pressure.

Model 2 has the worse results compares to the other two models: bigger prediction errors and the occurrence of offset. Although the simulated outputs remain constant when the input signals are constantly return to their initial conditions, the simulated output for upstream and downstream pressures do not returned to their initial condition. Inappropriately, the simulated outputs of Model 2 shift to the opposite directions of the real outputs.

Table 3.7 shows the best fit criterions of the models using APRBS response data. In line with the plots in Figure 3.23 and Figure 3.24, the highest best fit criterions are given by Model 3. The small best fit criterions of Model 1 are caused by the constant errors of prediction, hence the sum of absolute prediction error are increasing over time. The definition of the best fit criterion in (3.48) allows this value to be negative. As can be

seen, Model 2 has negative value of the best fit criterion for both upstream and downstream simulated outputs. The negative value of the best fit criterion means that the sum of prediction error is bigger than the deviation of measured output from its average value. One of the reasons is the validation data set was not preprocessed in the same way as the estimation data set [91], which in this case the input signals of identification data are different from the input signals of validation data.



**Figure 3.24:** The comparison of the real output from APRBS response data and the simulated outputs generated by Model 1, Model 2, and Model 3 for the downstream pressure.

**Table 3.7:** Performance comparison of Model 1, Model 2, and Model 3

Simulated output	Best fit criterion (%)		
	Model 1	Model 2	Model 3
PT202 (upstream pressure)	9.15	-241.38	68.48
PT212 (downstream pressure)	8.30	-171.47	75.07

The performance of Model 2 implies that the open-loop identification using PRBS input signals with simultaneous excitation for MIMO system is not a guaranteed to obtain a good model, especially when the interactions between the outputs or the internal variables in the plant are much more dominant or having the greater plant gains and the longer time constants rather than the responses from the input variables. In this case, the rapid frequency changes may have deteriorated some significant information about the plant dynamics.

Analyzing the performances of the three state-space models which are obtained using subspace system identification techniques with different approaches, it is found that the best open-loop performance is given by Model 3. It can be implied from the simulation plots and the best fit performance criterion that among the three obtained models, Model 3 is the most robust against the nonlinearities in the gaseous pilot plant.

There is also limitation in the development of an empirical model from input-output data using system identification techniques. As can be observed in Figure 3.23, all the developed models ever exceed 7 barg, while the actual maximum output of the gaseous pilot plant is 7 barg. However, it is difficult to avoid this error, since there is no model constraint considered in the construction of an empirical model using linear system identification techniques.

In the nature of interacting series processes, the interaction between the upstream and the downstream variables happen in both ways round. When the plant is strongly linear with some disturbances that can be approached with white noise signals, the empirical modeling using the model structure that taking into account of only the external inputs as the model inputs, such as what shown in (3.38), which can also be extended for any  $p$  inputs and  $m$  outputs, may an appropriate choice. However, it should be noted that such a model structure considers the interactions between upstream and downstream variables only as the additional linear combinations of the input variables. Suppose there is nonlinearity or un-periodic disturbance occurs in the upstream, it will results the simulation errors not only in the simulation of upstream output(s) but also in the simulation of downstream output(s), since in the plant the upstream variables will affect the downstream variables directly, and vice versa. When (3.40) is used as the model structure to predict the output of downstream variable, such nonlinearities in the upstream will be detected by the upstream sensor or transmitter, and then will be used as one of the

model inputs. In the same way, when (3.39) is used to predict the upstream variable, such nonlinearities will affect first to the downstream variable, then be detected by downstream sensor or transmitter, and further be used as one of the model inputs. Although it may resulting a small delays, the experiment results prove that this method can give the effective prediction and the robustness over an interacting series process, in this case the gaseous pilot plant, especially when the nonlinearities occur.

### 3.7 Summary

In this chapter, three practical approaches to construct a linear state-space model of an interacting series process, i.e the gaseous pilot plant, were presented. Three criterions were used to estimate the respective model order, while their performances were measured using the best fit criterion. The APRBS response data were used to perform the global comparison among the three models, and has shown that the model constructed from two MISO models gave the best performance. The technique to combine several MISO models into a single MIMO state-space model was also presented.

One of the important outcomes here is the approach that allows a designer to develop a state-space model of an interacting series process from input-output data using a linear system identification technique, with grey-box model structure which is constructed from several MISO models.

The work presented here improves the procedures required for developing a linear state-space model of an interacting series process. Its effectiveness and robustness were demonstrated in the open-loop validation using APRBS responses data.

Further important investigation of the proposed method is to evaluate its performance in real-time. In the subsequent chapter, its application as the internal model of a real-time model predictive control (MPC) would be presented.

## Chapter 4

### Real-Time Model Predictive Control of Interacting Series Process with Nonlinear Dynamics

A real-time implementation of a linear model predictive control (MPC) for controlling an interacting series process is presented in this chapter. An embedded model predictive controller consisting of the off-the-shelf components is implemented on a gaseous pilot plant that is used as the controlled process. The linear state-space model is used as the internal model of MPC, while the control actions are derived from the constrained optimization method, i.e, quadratic programming. Three linear models are tested and their effects to the controller's robustness are analyzed. To improve the controller's performances, four tuning parameters are investigated. A simple set of guidelines for tuning MPC controller is proposed in this work. The possibility to combine MPC and PID in the control system is also studied. Several issues in the real-time implementation of MPC controller are presented and discussed.

#### 4.1 Introduction

It is known that MPC is one of the most popular multivariable controls implemented in the process industries, and, promises several advantages, such as its algorithm can deal with multivariable case, provides the systematic way to the treatments of constraints and taking into account of actuator limitations, introduces feedforward control in a natural way to compensate the measurable disturbances. However, the computational algorithm of MPC is somehow more complex than the classical PID control. The controller output of MPC is also based on the prediction of process output generated by the internal model, which commonly is a kind of LTI models, even though it is well known that most of the real systems of interest are nonlinear, time-varying, may contain delays, and some variables or signals of central importance may not able to be measured directly. Moreover, MPC is a kind of discrete control systems, which produces the control signals in every sampling time. These conditions may presence some challenges in the practical implementation of MPC.

When it goes to real-time implementation, the hardware embedded MPC would be more appropriate to be used. Even though some applications are still prefer to use a dedicated



personal computer (PC) for implementing MPC, however, it seems to be less appropriate due to the possibility of user intervention. It is well known that in the practical process control, the control algorithms are commonly embedded in the control modules which are separated from the data recording and monitoring purposes. Hence, the embedded MPC is expected to function without user intervention, although it may require user interaction.

A class of series processes, as mentioned in the previous chapter, plays important roles in process industries since such types of interconnections are commonly found in their processing units. It was also mentioned that the interconnections of plant units may give rise to instability or oscillatory behavior, even when the individual processing units are stable. Moreover, due to the interactions between upstream and downstream units, series interconnections may introduce fundamental limitations in the achievable performance of control systems. Hence, it is necessary to study the possibility of the real-time implementation of multivariable controller, such as MPC, in the process plants that consist of series structures.

Many researches have extensively reported the implementation of various MPC in process plants [41, 55, 94-100], in which some of them contain the series structures [41, 94, 96, 98-100]. One of the remarkable papers of MPC implementation in the class of series processes is by Faanes and Skogestad [41] which present their study on the implementation of MPC for controlling noninteracting series processes with simulation study: pH neutralization performed in several steps (tanks). The model structure for noninteracting series processes has been developed in state-space form, and the study of control structures involves local control, where inputs to a unit are used to control outputs of the same unit, feedforward elements, and input resetting. One of the important results is that the multivariable controller for noninteracting series processes, in this case the original  $3 \times 3$  MPC controller used in the study, may rely strongly on feedforward control and thus be sensitive to model errors, including at steady-state conditions. Several options has been pointed out to improve the controller performance, such as examining the possibility of lacking integral action in the feedback part of the controller or perhaps modifying the process model.

Unlike the classical PID controller which basically has only three tuning parameters, i.e., proportional gain, integral time, and derivative time, MPC which has a more complex algorithm presents a more challenging options tuning strategy. MPC has a set of tuning

parameters, which can be used to fine-tune the closed-loop response for good performance and stability. Al-Ghazzawi et al. [27] point out that basically these parameters are adjusted via a trial and error procedure, which is a cumbersome task due to their overlapping effect and due to non-linearity brought by input constraints. However this procedure can be more systematic since there are several general tuning guidelines available in the literature. Shridhar and Cooper [26] has developed a novel tuning strategy for multivariable MPC, however, their approach is limited to unconstrained MPC, and it also requires representing the process by a first-order plus delay time (FOPDT) model, which may not work well for higher order and/or unstable process. Al-Ghazzawi et al. [27] presents the on-line tuning strategy for linear MPC algorithms based on the linear approximation between the closed-loop predicted output and the MPC tuning parameters. It is presented that by direct utilization of the sensitivity expressions for closed-loop response with respect to the MPC tuning parameters, new values of the tuning parameters can be found to steer the MPC feedback response inside predefined time-domain performance specifications. Trierweiler and Farina [28] present a novel tuning strategy for MIMO MPC. The proposed MPC tuning procedure is based on robust performance number (RPN), which can measure the difficulties of control problem. The authors, in agree with Wojsznis et al. [29], note that these tuning strategies have generally concentrated on distinct aspects, such as tuning for one of the criterion, e.g., controller robustness or performance, however, applying them in MPC implementation and operation has required sophisticated analysis tools and advanced knowledge of control approach. Wojsznis et al. [29] present a practical tuning approach, which features simple calculation of controller design parameters. Yet this calculation of tuning parameters seems to be suitable only for certain conditions. The experimental formula to define a penalty on move factor requires the presence of dead time in MPC scans, otherwise the calculation will results the same value of penalty on move factor for all kind of process. The authors also note that all tuning strategies mentioned above [26-29] are validated with simulations, which may produce different results when it goes to real-time applications.

Since the MPC computation algorithms are strongly rely on the internal model, which is well known that the linearized models are only valid for certain neighborhood of operating points, the use of MPC is also only appropriate for certain ranges. Although MPC promises several advantages, in the real process control system its presence cannot

simply eliminate PID. In practice, decentralized control structures, such as PID, are preferred in certain cases, in examples for ease of start-up [101, 102], bumpless automatic/manual transfer [102, 103], and fault tolerance in the event of actuator or sensor failures [102, 104, 105]. Hence, it is necessary to study the possibility of combining PID and MPC in the control system, so that they can be switched for their respective purposes.

This chapter presents a real-time implementation of an embedded MPC in an interacting series process, i.e., the gaseous pilot plant, where nonlinearities occur during plant operation. The study of tuning the model predictive controller is also presented in this chapter. The investigation of MPC tuning strategy in this work is carried out in real-time, instead of in simulation. Since there are several tuning guidelines of MPC available in the literature, which are validated using simulation studies, the purpose of tuning of the MPC in real-time would be to cover some aspects that perhaps cannot be seen in ideal environment (simulation). However, the objective of this study is not on finding a formula that produce exact values of the best tuning parameters, but to illustrate the effects of tuning several available parameters to the closed-loop responses of MPC in real-time conditions. Yet, since there is no “true” system that can really be captured by a model, it is believed that, in the practical applications, the best tuning strategy remains intuitive and iterative, so that it is the engineering skills and deep insights into the systems, including the control systems and the real systems of interest, that help to find the best tuning strategy. Since a linear model is commonly used as the internal model of MPC controller, and furthermore, the computational method of MPC is quite complex, the advantages of MPC are automatically followed by the number of disadvantages, so that, in certain purposes, the user may also prefer to use a more classical control system such as PID. In this chapter, the possibility to combine MPC and PID in a control system is also demonstrated, which may be able to bring it closer to the real applications where several purposes, such as safety or fault tolerance, are preferred to be handled by the more conventional controllers such as PID.

This chapter is organized as follows. Section 4.2 gives the methodology of real-time implementation of the model predictive controller presented in this chapter. Section 4.3 introduces the theories of model predictive control algorithm. Section 4.4 presents the overview of rapid prototyping of the embedded MPC used in this work. Section 4.5

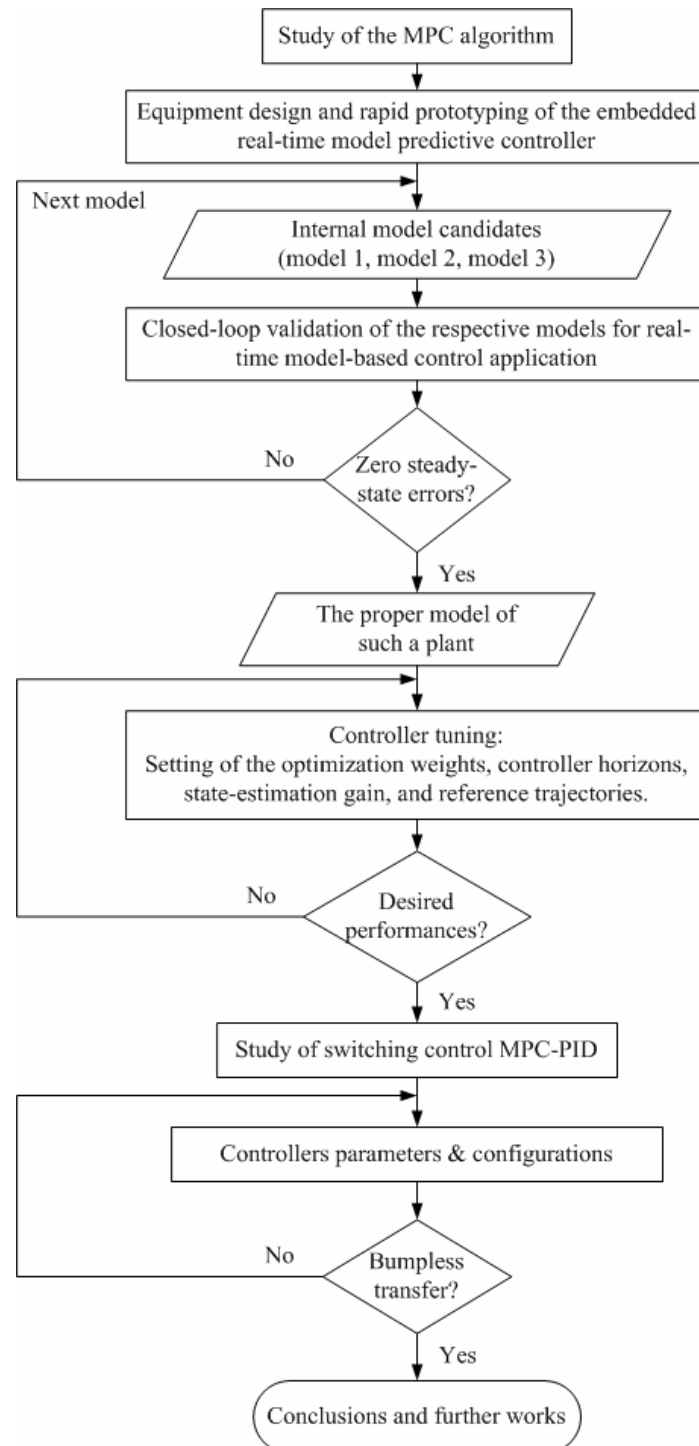
discusses the real-time implementation of model predictive control in the gaseous pilot plant. This section begins with the analysis of model mismatch in MPC, using the three models that have been obtained in previous chapter, and continues with fine-tuning the closed-loop responses of MPC. The discussion of MPC and PID control system alongside the possibility of their combination is given in section 4.6. Some discussions are presented in the last part of this section. Finally the summary of this chapter is given in section 4.7.

## 4.2 The Methodology of Real-Time Implementation of MPC

The flow diagram of the methodology used in the real-time implementation of model predictive control in an interacting series process, the gaseous pilot plant, that would be evolved in this research is given in Figure 4.1. As can be observed, it contains some iterative procedures.

The investigation would be begun with the study of the model predictive control algorithm, which is one of the most popular model-based controls in recent applications. MPC with constrained quadratic objective function would be considered in this work. To implement such an algorithm, experimental equipment would be designed, consisting of the off-the-shelf components and software tools. The next step would be to perform the closed-loop validation of the developed models in the previous chapter by using them as the internal model of the real-time model predictive controller for a given operating condition. The necessary condition is that the selected model has to be able to produce zero steady-state errors, which imply that such a model is robust against the plant's nonlinearities in the closed-loop control system. When the proper model is obtained, the study would then be followed by tuning the controller, so that the plant responses can shape as what it is desired. The tuning of MPC controller would be carried out in a heuristic way, since the standard tuning rules of MPC have not been reported in any literatures. This study would further be continued by examining the performance of PID and MPC in the gaseous pilot plant for a given operating point, and the possibility of MPC to work side-by-side with PID. As one of common issues in switching control systems, a bumpless transfer strategy in MPC would be examined, so that the controller would able to produce a smooth transition when it is switched from PID to MPC.

Details of the real-time implementation of model predictive control of an interacting series process which have been described above are presented in these subsequent sections.



**Figure 4.1:** The flow diagram of real-time model predictive control implementation.

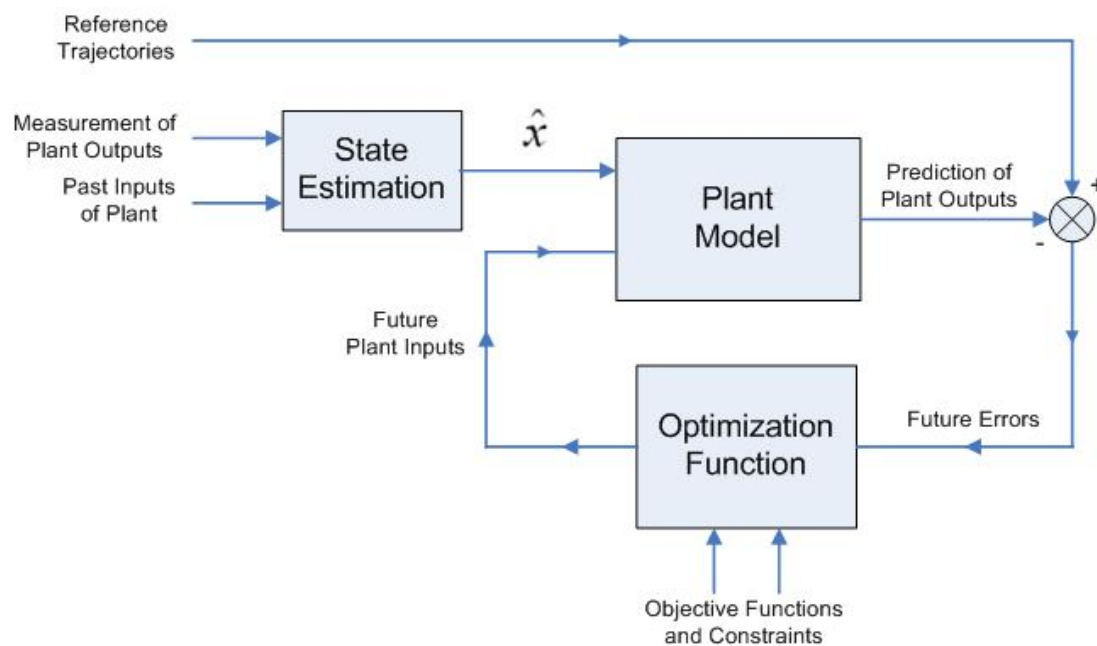
### 4.3 Model Predictive Control Theory

This section outlines the formulation of model predictive control algorithm. In general the MPC algorithm possesses three main elements, which are [68]:

- Prediction model.
- Objective function.
- Control law.

Different options can be chosen for each one of these elements giving rise to different algorithms. The descriptions of MPC algorithm in this section would be limited to the type of state-space plant model and the standard quadratic objective function. The derivation of MPC's control law is divided into unconstrained MPC problem and constrained MPC problem.

#### 4.3.1 Principle of model predictive control



**Figure 4.2:** Model predictive control illustration.

The basic concept of MPC is the utilization of long-range prediction from process output to minimize one or more objective functions, to produce an optimal control law. The prediction of process output for several steps ahead is derived using mathematical

equations that represent the model of controlled process. By using control law, the controller will compute the set of control actions for prediction of process output to reach the set of value that has been stated before. The output of control computation will be assigned to the process. The set of value that wanted to be reached by prediction of process output is called the reference trajectory [68, 74]. An illustration of the model predictive control algorithm is shown in Figure 4.2.

The MPC's control law is derived from minimizing one or more objective functions that generally can be written in the form below [50, 68]:

$$\min \mathbf{J} = \sum_{i=n}^p \mathbf{Q}(\mathbf{r}(k+i|k) - \hat{\mathbf{y}}(k+i|k))^2 + \sum_{l=0}^{m-1} \mathbf{R}(\Delta \mathbf{u}(k+l|k))^2, \quad (4.1)$$

subject to:

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max} \quad (4.2)$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \quad (4.3)$$

$$\hat{\mathbf{y}}_{\min} \leq \hat{\mathbf{y}} \leq \hat{\mathbf{y}}_{\max} \quad (4.4)$$

with  $\mathbf{r}$ ,  $\hat{\mathbf{y}}$ , and  $\Delta \mathbf{u}$  represent reference trajectory, predicted output, and the changes in the control signals, while  $n$ ,  $p$ , and  $m$  represent the length of minimum horizon, prediction horizon, and control horizon, consecutively.  $\mathbf{Q}$  and  $\mathbf{R}$  are weighting matrices that set relative degrees of penalty on the deviations of the predicted plant outputs from the reference trajectory and the changes in the future control inputs.  $\mathbf{Q}$  and  $\mathbf{R}$  are symmetric, positive semi-definite, and usually diagonal, which have the dimension  $n_y \times n_y$  and  $n_u \times n_u$  respectively.

The inequalities in (4.2) and (4.3) are constraints on the magnitude of and change in the control inputs, which may come from physical limitation of actuators. (4.4) is constraints on the magnitude of plant outputs, which may be dictated by certain economical objectives or requirements on the dynamic response of the plant. If a linear model is used and the inequalities in (4.2) until (4.4) are linear, obtaining the solution of (4.1) that subject to (4.2) until (4.4) can be addressed as solving a quadratic programming (QP) problem. If the Hessian matrix of the objective function is positive semi-definite, it is a convex QP problem which has a global minimum. If the Hessian matrix is positive definite, it will have a unique global minimum [106].

The internal model used in MPC can be:

- a. A step response model or finite impulse response model.
- b. A discrete transfer function.
- c. A state-space model.

In general, these three forms of model is convertible to each other. In this research, the state-space model is used. The general state-space model of discrete linear time invariant system can be formed in the following equation [82]:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k),\end{aligned}\tag{4.5}$$

$\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are the state coefficient matrices, the dimension of which are  $n_x \times n_x$ ,  $n_x \times n_u$ ,  $n_y \times n_x$ , and  $n_y \times n_u$  respectively.

The reference trajectory can be a step shape signal, ramp, or exponential curve defined by the user. When there is a change in the setpoint or deviates from the setpoint due to some disturbance, it may be necessary to bring the output back to the setpoint as quickly as possible. However, in some cases the quick response may cause too large control actions and oscillations in the output. Therefore the reference trajectory can be modified so that the controller response to the setpoint change or the disturbance is less aggressive.

#### 4.3.2 State estimation

Besides the information about plant inputs and outputs, using a state-space plant model in (4.5) for prediction requires the information about the state at the current time  $k$ ,  $\mathbf{x}(k)$ . However, it is often not practical to directly measure all the state variables of the system. In addition, it is common in state-space modeling that the state does not necessarily has a direct physical interpretation but they have a conceptual relevance [45, 106]. Therefore, the state estimator (observer) should be used to obtain the estimated state,  $\hat{\mathbf{x}}(k|k)$ , from the plant inputs and measurements of plant outputs.

The state estimation of plant is obtained using the following equations [107]:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}[\mathbf{y}(k) - \hat{\mathbf{y}}(k|k-1)];\tag{4.6}$$

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k) + \mathbf{B}\mathbf{u}(k);\tag{4.7}$$

$$\hat{\mathbf{y}}(k|k-1) = \mathbf{C}\hat{\mathbf{x}}(k|k-1) + \mathbf{D}\mathbf{u}(k),\tag{4.8}$$

with  $\hat{\mathbf{x}}$  is estimated state and  $\mathbf{K}$  is estimator gain.



The estimator gain  $\mathbf{K}$  can be determined by pole placement techniques, but for MIMO systems,  $\mathbf{K}$  will be a matrix, which is more easily and practically determined by the optimal estimation method. The optimal estimator is also known as Kalman filter [37, 69, 106].

In the optimal estimation, it is assumed that the system is subject to the plant noise  $\mathbf{w}(k)$  and measurement noise  $\mathbf{v}(k)$ , with  $\mathbf{\Lambda}$  is the coefficient matrix for plant noise, and mathematically can be expressed as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{\Lambda}\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{v}(k).\end{aligned}\tag{4.9}$$

Both  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  are assumed to be white noise with zero mean and impulse autocorrelation, that is,

$$\begin{aligned}\mathbf{E}[\mathbf{w}(k)] &= \mathbf{0} ; \\ \mathbf{E}[\mathbf{v}(k)] &= \mathbf{0} ,\end{aligned}\tag{4.10}$$

$$\mathbf{E}[\mathbf{w}(i)\mathbf{w}(j)^T] = \begin{cases} \mathbf{0} & \text{if } i \neq j, \\ \mathbf{R}_w & \text{if } i = j, \end{cases}\tag{4.11}$$

$$\mathbf{E}[\mathbf{v}(i)\mathbf{v}(j)^T] = \begin{cases} \mathbf{0} & \text{if } i \neq j, \\ \mathbf{R}_v & \text{if } i = j, \end{cases}\tag{4.12}$$

$\mathbf{w}(k)$  and  $\mathbf{v}(k)$  are uncorrelated.  $\mathbf{R}_w$  and  $\mathbf{R}_v$  are covariances reflecting the noise level, and for a MIMO system,  $\mathbf{R}_w$  and  $\mathbf{R}_v$  are in matrix forms.

The steady-state optimal estimator gain can be found from (4.13), and  $\mathbf{P}$  is the solution of the algebraic Ricatti equation in (4.14), which are

$$\mathbf{K} = \mathbf{P}\mathbf{C}^T (\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}_v)^{-1},\tag{4.13}$$

$$\mathbf{P} = \mathbf{A}\mathbf{P}\mathbf{A} - \mathbf{A}\mathbf{P}\mathbf{C}^T (\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}_v)^{-1} \mathbf{C}\mathbf{P}\mathbf{A}^T + \mathbf{\Lambda}\mathbf{R}_w\mathbf{\Lambda}^T.\tag{4.14}$$

Different  $\mathbf{K}$  can be found based on the knowledge of the noise levels  $\mathbf{R}_w$  and  $\mathbf{R}_v$  [106].

### 4.3.3 Prediction of process output

Using the state-space model in (4.5), the prediction equation from the process can be stated as the function from future and current inputs with the estimated state variable,  $\hat{\mathbf{x}}$ , which can be formulated in the matrices equation as shown below [68]:

$$\begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \vdots \\ \hat{\mathbf{y}}(k+p|k) \end{bmatrix} = \mathbf{F}_x \hat{\mathbf{x}}(k|k) + \mathbf{F}_u \mathbf{u}(k|k-1) + \mathbf{G} \begin{bmatrix} \Delta \mathbf{u}(k|k) \\ \vdots \\ \Delta \mathbf{u}(k+p-1|k) \end{bmatrix}, \quad (4.15)$$

with

$$\mathbf{F}_x = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^p \end{bmatrix} \in \mathbb{R}^{pn_y \times n_x}; \quad (4.16)$$

$$\mathbf{F}_u = \begin{bmatrix} \mathbf{CB} \\ \mathbf{CB} + \mathbf{CAB} \\ \vdots \\ \sum_{h=0}^{p-1} \mathbf{CA}^h \mathbf{B} \end{bmatrix} \in \mathbb{R}^{pn_y \times n_u}; \quad (4.17)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} + \mathbf{CAB} & \mathbf{CB} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{h=0}^{p-1} \mathbf{CA}^h \mathbf{B} & \sum_{h=0}^{p-2} \mathbf{CA}^h \mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \in \mathbb{R}^{pn_y \times pn_u}. \quad (4.18)$$

The set of  $\hat{\mathbf{y}}(k+p|k)$  for  $p = 0, 1, \dots, p$  can be rewritten as

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f}, \quad (4.19)$$

with  $\mathbf{f} = \mathbf{F}_x \hat{\mathbf{x}}(k|k) + \mathbf{F}_u \mathbf{u}(k|k-1)$  represents the free responses of the system.

### 4.3.4 Unconstrained MPC

If there are no constraints on the control inputs and plant outputs, the solution of (4.1) can be simply found at the global minimum of objective function.

The reference trajectory and the diagonal weighting matrices can be formulated as

$$\mathbf{r} = [\mathbf{r}(k+1|k) \quad \mathbf{r}(k+2|k) \quad \cdots \quad \mathbf{r}(k+p|k)]^T, \quad (4.20)$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}(1) & \mathbf{0} & \cdots & \mathbf{0} \\ 0 & \mathbf{Q}(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}(p) \end{bmatrix}, \quad (4.21)$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}(p) \end{bmatrix}. \quad (4.22)$$

The objective function in (4.1) can be reformed in the matrices equation as

$$\begin{aligned} \mathbf{J} = & \begin{bmatrix} \mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k) \\ \mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k) \\ \vdots \\ \mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k) \end{bmatrix}^T \mathbf{Q} \begin{bmatrix} \mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k) \\ \mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k) \\ \vdots \\ \mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k) \end{bmatrix} \\ & + \begin{bmatrix} \Delta \mathbf{u}(k|k) \\ \Delta \mathbf{u}(k+1|k) \\ \vdots \\ \Delta \mathbf{u}(k+m-1|k) \end{bmatrix}^T \mathbf{R} \begin{bmatrix} \Delta \mathbf{u}(k|k) \\ \Delta \mathbf{u}(k+1|k) \\ \vdots \\ \Delta \mathbf{u}(k+m-1|k) \end{bmatrix}, \end{aligned} \quad (4.23)$$

or simply

$$\begin{aligned} \mathbf{J} &= (\mathbf{r} - \mathbf{G}\Delta \mathbf{u} - \mathbf{f})^T \mathbf{Q} (\mathbf{r} - \mathbf{G}\Delta \mathbf{u} - \mathbf{f}) + \Delta \mathbf{u}^T \mathbf{R} \Delta \mathbf{u} \\ &= \hat{\mathbf{\varepsilon}}^T \mathbf{Q} \hat{\mathbf{\varepsilon}} + \Delta \mathbf{u}^T \mathbf{R} \Delta \mathbf{u}, \end{aligned} \quad (4.24)$$

with  $\hat{\mathbf{\varepsilon}} = \mathbf{r} - \hat{\mathbf{y}} = \mathbf{r} - \mathbf{G}\Delta \mathbf{u} - \mathbf{f}$  represents closed-loop prediction error based on future and current control actions.

When the first derivative of a function is equal to zero, then at that value, the respective function will on its local minimum, local maximum, or turning point. Mathematically, it can be expressed as

$$\mathbf{g} = \frac{\partial \mathbf{J}}{\partial \Delta \mathbf{u}} = 0. \quad (4.25)$$

The second derivative should be examined to check whether a function on its local minimum, local maximum, or turning point when its first derivative is equal to zero. If the second derivative is positive, then for  $\mathbf{g} = 0$ , a function will be on its local minimum.

In addition, the local minimum of the function will be global minimum when its second derivative is positive definite, which is

$$\mathbf{g}' = \frac{\partial^2 \mathbf{J}}{\partial \Delta \mathbf{u}^2} > 0. \quad (4.26)$$

The matrix  $\mathbf{H}$  is said to be positive definite if satisfy [108]:

$$\mathbf{a}^T \mathbf{H} \mathbf{a} > 0, \quad (4.27)$$

for all nonzero complex vector  $\mathbf{a}$ .

The first derivative of objective function in (4.24) is

$$\begin{aligned} \mathbf{g} = \frac{\partial \mathbf{J}}{\partial \Delta \mathbf{u}} &= -2\mathbf{G}^T \mathbf{Q}((\mathbf{r} - \mathbf{f}) - \mathbf{G} \Delta \mathbf{u})^T + 2\mathbf{R} \Delta \mathbf{u} \\ &= 2((\mathbf{G}^T \mathbf{Q} \mathbf{G}) \Delta \mathbf{u} - \mathbf{G}^T \mathbf{Q}(\mathbf{r} - \mathbf{f})) . \end{aligned} \quad (4.28)$$

While the second derivative of that objective function is

$$\mathbf{g}' = \frac{\partial^2 \mathbf{J}}{\partial \Delta \mathbf{u}^2} = 2(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R}). \quad (4.29)$$

It can be seen that the second derivative of the objective function in (4.29) is not depend on  $\Delta \mathbf{u}$ . Since  $\mathbf{Q}$  and  $\mathbf{R}$  are symmetric and positive semi-definite, while all elements of  $\mathbf{G}$  is real, then  $\mathbf{g}'$  is positive definite.

The global minimum of objective function in (4.24) can be obtained by equating the first derivative in (4.28) to zero as

$$2((\mathbf{G}^T \mathbf{Q} \mathbf{G}) \Delta \mathbf{u} - \mathbf{G}^T \mathbf{Q}(\mathbf{r} - \mathbf{f})) = 0. \quad (4.30)$$

The set of the changes in the control signals that minimizes MPC's objective function for unconstrained case is given in the following equation:

$$\Delta \mathbf{u} = (\mathbf{G}^T \mathbf{Q} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{Q}(\mathbf{r} - \mathbf{f}). \quad (4.31)$$

And the relation between  $u$  and  $\Delta u$  can be expressed as

$$\mathbf{u}(k+i | k) = \Delta \mathbf{u}(k+i | k) + \mathbf{u}(k+i-1 | k). \quad (4.32)$$

#### 4.3.5 Constrained MPC

In the real practical applications, there are always some constraints put on the control input and/or plant output. The constraints can be categorized in terms of the source. An actuator has certain limit-to-limit travel time that restricts the change in the actuator position to a limited rate. This restriction is then reflected by a constraint on the changes in the control inputs of MPC. For instance, if a control valve has a travel time of 10 seconds from fully open to fully close and MPC has a control interval of 1 second, then the maximal change in its control inputs will be 10% of the full range. Another constraint on the actuator is its maximum and minimum limit. For instance, a control valve may have a range of 0% - 100% of opening. However, due to some reasons, e.g., process safety, it may not be allowed the control valve to be fully open or fully closed. The constraints on the plant outputs can serve two purposes. One is to affect the transient behavior of the output. If the lower and upper bounds upon the plant outputs are put all over the prediction horizon, MPC will generate the control inputs to restrict the overshoot and undershoot of the output within the bounds. However, the actual control quality is affected by the accuracy of the internal model. The model mismatch may cause the actual overshoot or undershoot beyond the limits set in MPC. The other purpose of the output constraints is to control the steady-state output of plant. In some cases, a tight control on the plant outputs is unnecessary or unrealistic, so it is acceptable or more practical to control the output within the range. Using MPC, such non-tight control strategy can be implemented in a straightforward way. The weighting element of plant outputs  $\mathbf{Q}$  can be set to a small number, which tells the MPC controller that the plant outputs deviations will not be seriously penalized, on the other hand, the upper and lower limits of the plant outputs should be set in the constraint, which instructs the MPC to generate control inputs to consider the limits of plant outputs. Controllers in the real applications must be capable of handling constraints in some way. Conventional control strategy uses *ad hoc* controllers to deal with different constraints, while MPC is able to handle constraints explicitly [106].

The constraints can also be divided into hard and soft constraints. A hard constraint cannot be violated when solving the optimization problem, while violations of a soft constraint can be allowed but are penalized in the objective function. In MPC, the hard constraints may come from the physical limitations of a system which cannot be compromised, such as the maximal pressure inside the vessel, the maximum and

minimum of opening valve, or the largest change in the control valve in one control interval. Violations of such hard constraints can degrade the performance of an MPC controller because the controller is not aware that the control input is unrealistic [106]. The soft constraints are adopted to maintain feasibility of an optimization. Soft constraints are obtained by softening hard constraints. To soften the constraints, slack variables could be added to the upper and lower bounds of the hard constraints, which relax the bounds. Furthermore, the objective function should be modified so that it includes a term to penalize the magnitude of slack variables. With the use of slack variables, violations of the original hard constraints are allowed but penalized [106]. However, in this study, only hard constraints of MPC will be examined.

It is necessary to reform all the constraints in (4.2) until (4.4) in terms of either  $\Delta \mathbf{u}$  or  $\mathbf{u}$ . Furthermore, in order to be solved by the optimization algorithm, the constraints are necessary to be formulated in the matrices form. This formulation is shown in (4.33) until (4.48).

Let  $\mathbf{u}$  is chosen as the output of optimization, then the relation between  $\Delta \mathbf{u}$  and  $\mathbf{u}$  can be written as

$$\Delta \mathbf{u} = \Xi \mathbf{u} - \mathbf{f}_1, \quad (4.33)$$

with

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta \mathbf{u}(k) \\ \Delta \mathbf{u}(k+1) \\ \Delta \mathbf{u}(k+2) \\ \vdots \\ \Delta \mathbf{u}(k+m) \end{bmatrix}; \quad (4.34)$$

$$\Xi = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{1} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{1} & \mathbf{1} \end{bmatrix}; \quad (4.35)$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \mathbf{u}(k+2) \\ \vdots \\ \mathbf{u}(k+m) \end{bmatrix}; \quad (4.36)$$

$$\mathbf{f}_1 = \begin{bmatrix} \mathbf{u}(k-1) \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \quad (4.37)$$

First, the constraint of the change in control signal is formulated. Suppose that  $\Delta \mathbf{u}_{\min}$  and  $\Delta \mathbf{u}_{\max}$  are the constraint of  $\Delta \mathbf{u}$ , then by substituting (4.33) into (4.2) the inequality becomes

$$\Delta \mathbf{u}_{\min} + \mathbf{f}_1 \leq \Xi \mathbf{u} \leq \Delta \mathbf{u}_{\max} + \mathbf{f}_1, \quad (4.38)$$

which implies that

$$\Xi \mathbf{u} \leq \Delta \mathbf{u}_{\max} + \mathbf{f}_1 \text{ and } -\Xi \mathbf{u} \leq -(\Delta \mathbf{u}_{\min} + \mathbf{f}_1), \quad (4.39)$$

or in the matrix form, it can be expressed as

$$\begin{bmatrix} \Xi \\ -\Xi \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} \Delta \mathbf{u}_{\max} + \mathbf{f}_1 \\ -(\Delta \mathbf{u}_{\min} + \mathbf{f}_1) \end{bmatrix}. \quad (4.40)$$

The constraint of the magnitude of control signal expressed in (4.3) can be reformed as

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} \mathbf{u}_{\max} \\ -\mathbf{u}_{\min} \end{bmatrix}, \quad (4.41)$$

with  $\mathbf{I}$  is the identity matrix.

The prediction of plant output can be reformed by substituting (4.33) to (4.19) as

$$\hat{\mathbf{y}} = \mathbf{F}\mathbf{u} + \mathbf{f}_2, \quad (4.42)$$

with

$$\mathbf{F} = \mathbf{G}\Xi \text{ and } \mathbf{f}_2 = \mathbf{f} - \mathbf{G}\mathbf{f}_1. \quad (4.43)$$

Using (4.42), the constraint of plant output in (4.4) can be written as

$$\hat{\mathbf{y}}_{\min} \leq \mathbf{F}\mathbf{u} + \mathbf{f}_2 \leq \hat{\mathbf{y}}_{\max}. \quad (4.44)$$

The inequality in (4.44) implies

$$-\mathbf{F}\mathbf{u} \leq -(\hat{\mathbf{y}}_{\min} - \mathbf{f}_2) \text{ and } \mathbf{F}\mathbf{u} \leq \hat{\mathbf{y}}_{\max} - \mathbf{f}_2, \quad (4.45)$$

or in the matrix form become

$$\begin{bmatrix} \mathbf{F} \\ -\mathbf{F} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} \hat{\mathbf{y}}_{\max} - \mathbf{f}_2 \\ -(\hat{\mathbf{y}}_{\min} - \mathbf{f}_2) \end{bmatrix}. \quad (4.46)$$

By combining (4.40), (4.41), and (4.46) all the constraints can be expressed as

$$\begin{bmatrix} \Xi \\ -\Xi \\ \mathbf{I} \\ -\mathbf{I} \\ \mathbf{F} \\ -\mathbf{F} \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} \Delta \mathbf{u}_{\max} + \mathbf{f}_1 \\ -(\Delta \mathbf{u}_{\min} + \mathbf{f}_1) \\ \mathbf{u}_{\max} \\ \mathbf{u}_{\min} \\ \hat{\mathbf{y}}_{\max} - \mathbf{f}_2 \\ -(\hat{\mathbf{y}}_{\min} - \mathbf{f}_2) \end{bmatrix}, \quad (4.47)$$

or simply

$$\Psi \mathbf{u} \leq \psi. \quad (4.48)$$

#### 4.3.6 Quadratic programming problem

Unlike the unconstrained quadratic optimization problem, a convex QP problem generally has no closed-form solution. For unconstrained MPC problem, the solution can be simply obtained by equating its first derivative to zero as in (4.31), while for constrained problem, the presence of constraints variable make it impossible to find the solution in such way. A numerical method has to be employed to find the solution in an iterative way. There are many algorithms for solving a convex QP problem which includes pivoting algorithms, reduced gradient algorithms, active set methods, and interior point methods [106]. This study uses the QP solver – *qpdartz* in *MATLAB*<sup>®</sup> *Model Predictive Control Toolbox*, which adopts a Dantzig-Wolfe's active set method. The concept can be briefly outlined as below [109]:

The general form of a convex QP problem for is

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{2} \boldsymbol{\theta}^T \boldsymbol{\Phi} \boldsymbol{\theta} + \boldsymbol{\xi}^T \boldsymbol{\theta} \text{ subject to } \boldsymbol{\Omega}^T \boldsymbol{\theta} \geq \boldsymbol{\chi}, \quad \boldsymbol{\theta} \geq \mathbf{0}. \quad (4.49)$$

Introducing multipliers  $\boldsymbol{\lambda}$  for the constraints  $\boldsymbol{\Omega}^T \boldsymbol{\theta} \geq \boldsymbol{\chi}$  and  $\boldsymbol{\pi}$  for the bounds  $\boldsymbol{\theta} \geq \mathbf{0}$  gives the Lagrangian function:



$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\pi}) = \frac{1}{2} \boldsymbol{\theta}^T \boldsymbol{\Phi} \boldsymbol{\theta} + \boldsymbol{\xi}^T \boldsymbol{\theta} - \boldsymbol{\lambda}^T (\boldsymbol{\Omega}^T \boldsymbol{\theta} - \boldsymbol{\chi}) - \boldsymbol{\pi}^T \boldsymbol{\theta} . \quad (4.50)$$

For a convex QP problem, the necessary and sufficient conditions for  $x$  being the optimal are *Kuhn-Tucker* (KT) conditions [106, 109]. Defining slack variables  $\boldsymbol{\kappa} = \boldsymbol{\Omega}^T \boldsymbol{\theta} - \boldsymbol{\chi}$ , the KT conditions for (4.50) are

$$\boldsymbol{\pi} - \boldsymbol{\Omega} \boldsymbol{\theta} + \boldsymbol{\Omega} \boldsymbol{\lambda} = \boldsymbol{\xi} ; \quad (4.51)$$

$$\boldsymbol{\kappa} - \boldsymbol{\Omega}^T \boldsymbol{\theta} = -\boldsymbol{\chi} ; \quad (4.52)$$

$$\boldsymbol{\pi}, \boldsymbol{\kappa}, \boldsymbol{\theta}, \boldsymbol{\lambda} \geq \mathbf{0} ; \quad (4.53)$$

$$\boldsymbol{\pi}^T \boldsymbol{\theta} = \mathbf{0} ; \quad (4.54)$$

$$\boldsymbol{\kappa}^T \boldsymbol{\lambda} = \mathbf{0} . \quad (4.55)$$

The method assumes that  $\boldsymbol{\Phi}$  is positive definite: which is a sufficient condition that there is any solution for (4.51) until (4.55) to solve (4.49). The Dantzig-Wolfe method solves (4.51) until (4.55) in the iterative way to find  $\boldsymbol{\pi}$ ,  $\boldsymbol{\kappa}$ ,  $\boldsymbol{\theta}$ , and  $\boldsymbol{\lambda}$  that minimize the objective function in (4.49).

#### 4.3.7 Quadratic programming as a control law for constrained MPC

To obtain MPC's optimal control actions using QP, the objective function in (4.24) subject to the constraints in (4.47) has to be formed into the general notation of QP as in (4.49).

By substituting (4.42) and (4.33) into (4.24), the MPC's objective function in (4.1) can be written as

$$\begin{aligned} \mathbf{J}(\mathbf{u}) &= (\mathbf{F}\mathbf{u} + \mathbf{f}_2 - \mathbf{r})^T \mathbf{Q} (\mathbf{F}\mathbf{u} + \mathbf{f}_2 - \mathbf{r}) + (\boldsymbol{\Xi}\mathbf{u} - \mathbf{f}_1)^T \mathbf{R} (\boldsymbol{\Xi}\mathbf{u} - \mathbf{f}_1) \\ &= \mathbf{u}^T (\mathbf{F}^T \mathbf{Q} \mathbf{F} + \boldsymbol{\Xi}^T \mathbf{R} \boldsymbol{\Xi}) \mathbf{u} + 2 \left[ (\mathbf{f}_2 - \mathbf{r})^T \mathbf{Q} \mathbf{F} - \mathbf{f}_1^T \mathbf{R} \boldsymbol{\Xi} \right] \mathbf{u} \\ &\quad + (\mathbf{f}_2 - \mathbf{r})^T \mathbf{Q} (\mathbf{f}_2 - \mathbf{r}) + \mathbf{f}_1^T \mathbf{R} \mathbf{f}_1 . \end{aligned} \quad (4.56)$$

Using (4.56), the objective function in (4.1) can be reformed into the general notation of QP as follow:

$$\underset{\mathbf{u}}{\text{minimize}} \quad \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{c}^T \mathbf{u} + \mathbf{f}_c , \quad (4.57)$$

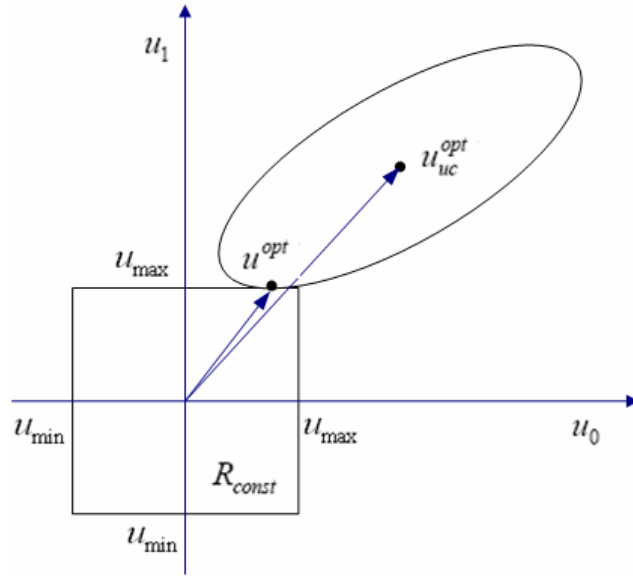
subject to inequality constraints  $\boldsymbol{\Psi} \mathbf{u} \leq \boldsymbol{\psi}$  and  $\mathbf{u} \geq \mathbf{0}$ , with

$$\mathbf{H} = 2(\mathbf{F}^T \mathbf{Q} \mathbf{F} + \mathbf{\Xi}^T \mathbf{R} \mathbf{\Xi}), \quad (4.58)$$

$$\mathbf{c}^T = 2[(\mathbf{f}_2 - \mathbf{r})^T \mathbf{Q} \mathbf{F} - \mathbf{f}_1^T \mathbf{R} \mathbf{\Xi}], \quad (4.59)$$

$$\mathbf{f}_c = (\mathbf{f}_2 - \mathbf{r})^T \mathbf{Q} (\mathbf{f}_2 - \mathbf{r}) + \mathbf{f}_1^T \mathbf{R} \mathbf{f}_1. \quad (4.60)$$

The matrix  $\mathbf{H}$  is a symmetric matrix which is indicated by  $\mathbf{H} = \mathbf{H}^T$ , and also a positive definite matrix which satisfy the condition in (4.27). These conditions imply that the optimization problem can be solved using QP method, and the solution of the optimization variable  $\mathbf{u}$  can be used as the optimal control actions produced by the MPC controller.



**Figure 4.3:** Geometric of 2-dimensions quadratic programming illustration.

The quadratic programming notation in (4.57) has a nice geometric interpretation in the  $\mathbf{u}$ -space. For example, let  $\mathbf{u} = [u(k|k) \ u(k+1|k)]$ ,  $u_{\min} \leq u \leq u_{\max}$ , and  $\mathbf{R}_{const}$  is the square plane  $[u_{\min}, u_{\max}] \times [u_{\min}, u_{\max}] \subset \mathbb{R}^2$ . The following equation

$$\frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{c}^T \mathbf{u} + \mathbf{f}_c = 0 \quad (4.61)$$

defines ellipsoids in  $\mathbb{R}^2$  centered at  $\mathbf{u} = -\mathbf{H}^{-1} \mathbf{c}$  which is the optimal solutions for unconstrained optimization problem ( $\mathbf{u}_{uc}^{opt}$ ). Then the optimization problem in (4.57) with these conditions can be regarded as the problem of finding the smallest ellipsoid that

intersects  $\mathbf{R}_{const}$ , and  $\mathbf{u}^{opt}$  as the solution is the point of intersection [110]. This geometric is illustrated in Figure 4.3.

#### 4.4 Real-Time Controller

The planning of control system commonly requires a series of steps which, depending on the complexity of the process, begin with modeling the dynamic system to be controlled, followed by identification of parameters associated with the process, analysis and validation of the model and the parameters obtained, definition and the type of controller, and finally with the real-time implementation of the control system as a whole. In particular, accounting for some of the implementation details such as the resolution of a fixed-point microprocessor that will be used may affect the originally designed controller model and require it to be modified [23, 111, 112].

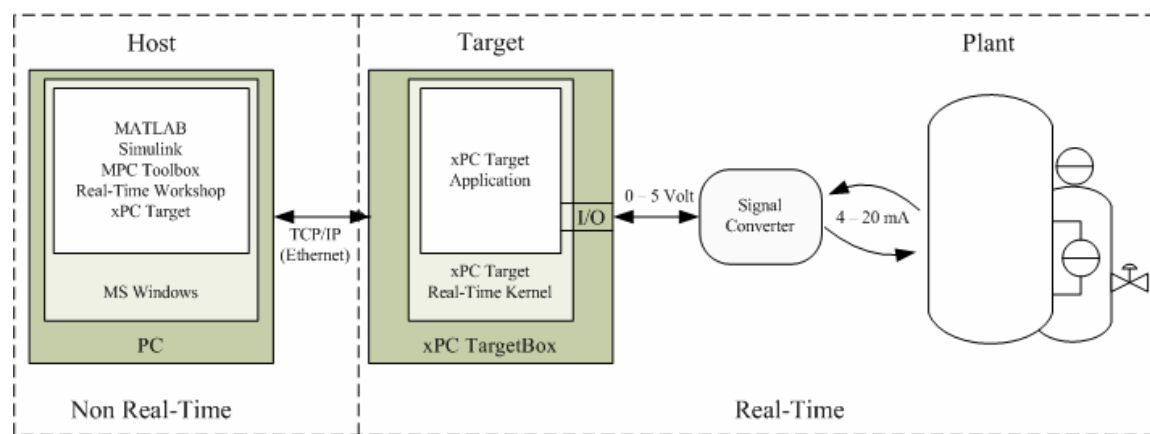
A rapid prototype is a quick way to validate the controller code by executing it with the actual plant, sensors and actuators, the plant model, or any combination of these components [111, 112]. In scientific research and education, where a system is rarely taken into production, rapid prototyping serves an important purpose. In industry, rapid prototyping allows testing of a partial design before expending the effort to include robustness measures and optimizing the design [112].

A significant advance came with the commercialization and distribution of certain simulation tools which markedly simplified the whole set of tasks associated with the simulation phase for planning controllers. One of the commercially available systems that widely used in the dynamic system analysis, simulation, and control area is MATLAB/Simulink by The Math Works, Inc., which recently has released a toolbox called xPC Target. The xPC Target provides the tools for the automatic generation of real-time code from Simulink models, without the user having to write low-level code and allows the user access to input/output (I/O) data directly from a compatible data acquisition card [112-115].

This section presents the rapid prototyping of a linear MPC based on MATLAB/Simulink environment. The MPC algorithm is carried out using the MPC Toolbox [37], which have been used by many researchers to implement the model predictive control algorithm, such as [107, 116-121]. Some developments and features that may be beneficial in its further applications are also reported.

#### 4.4.1 System description

The overall host-target real-time control system architecture is shown in Figure 4.4. The host PC is a general-purpose PC which is used to design the control configurations in the offline mode. The host PC also supervises and monitors the xPC TargetBox in real-time application. The xPC TargetBox is a hardware which is used for prototyping the control systems. It is a combination of xPC Target software and a Pentium-based computer in a rugged enclosure that is suitable for industrial environments. The microprocessor can be augmented with a number of I/O configurations that are commonly required for control applications, such as timers, counters, analog-to-digital (A/D) and digital-to-analog (D/A) converters, pulse-width modulation (PWM), digital I/O, and controller area network (CAN) bus. The xPC TargetBox systems can achieve sample rates approaching 60 kHz, however, it depends on the complexity of the program.



**Figure 4.4:** The real-time control system architecture.

On the host PC (which runs MATLAB, Simulink, Real-Time Workshop, MPC Toolbox, and xPC Target), xPC Target works with the code generated from the Simulink application and a C-compiler to build the real-time target application. The target application can run in real-time on a xPC TargetBox once it is downloaded from the host PC.

The target hardware is booted from a real-time kernel in xPC Target. However, the xPC Target kernel needs the PC basic input/output system (BIOS) because when the target PC boots and the BIOS is loaded, the BIOS prepares the target PC environment for running and then starts the kernel.

The kernel initiates the communication of host-target, activates the application loader, and waits for the target application to be downloaded from the host PC. The communication of host-target occurs through TCP/IP communication protocols using Ethernet. Once the target application has been downloaded to the xPC TargetBox, it can be monitored and supervised from the host PC.



**Figure 4.5:** Photograph of the xPC TargetBox with the I/O module (left) and the host PC (right).

Voltage-to-current and current-to-voltage converters are used, so that the I/O module of xPC TargetBox, which has the range of 0 – 5 Volt, able to communicate with the plant instrumentations which have 4 – 20 mA industrial standard signal.

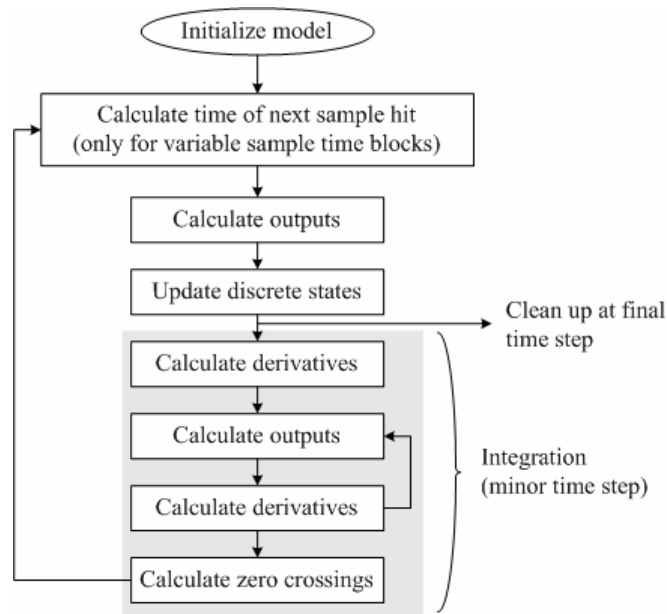
Figure 4.5 shows the photographs of the xPC TargetBox with its I/O module which are used to perform the real-time control, and the host PC which is used to design and simulate the controller in offline mode, and also to monitor and supervise the controller in real-time.

#### **4.4.2 Generating embedded code and application execution**

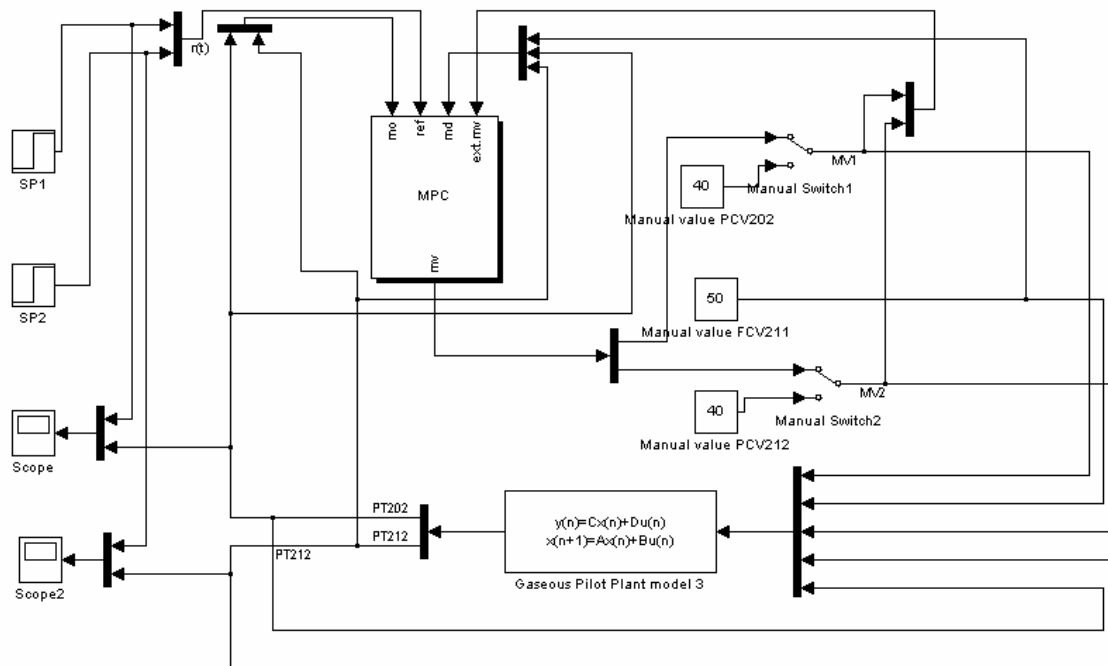
Basically, there are two steps to build a real-time controller using MATLAB, Simulink, and xPC Target. The first is to design and evaluate the control algorithm in offline mode and the second is to implement the designed controller into a device which able to perform the real-time control tasks. Brief overviews of these steps are outlined as follows:

#### 4.4.2.1 Simulink simulation steps

A typical Simulink block consists of inputs, states, and outputs. The outputs are a function of the sample time, the inputs, and the block states. During simulation, the model execution follows a series of steps as shown in Figure 4.6 [112].



**Figure 4.6:** Steps in a Simulink simulation.



**Figure 4.7:** Simulink diagram for offline simulation of MPC.

The simulation begins with the initialization of the model, which includes: incorporates library blocks into the model, propagation of signal widths, data types, and sample times; evaluates block parameters; determines block execution order; and allocates memory. Simulink then enters a simulation loop. Each pass through the loop is referred to as a simulation step. During each simulation step, Simulink executes each of the model blocks in the order determined during initialization. For each block, Simulink invokes functions that compute the values of the block states, the derivatives, and the outputs for the current sample time. The simulation is then incremented to the next step. This process continues until the simulation is stopped.

The model predictive control algorithm is first simulated in Simulink using MPC Toolbox. The Simulink diagram of this simulation is shown in Figure 4.7. The purpose of this simulation is to investigate the controller behaviors in offline mode. The plant model used to simulate the behavior of gaseous pilot plant is equal with the internal model of controller, resulting that the controller able to bring and maintain the process variables on the given set-points. the simulation of the model predictive control algorithm using MPC Toolbox has been reported by authors in [122].

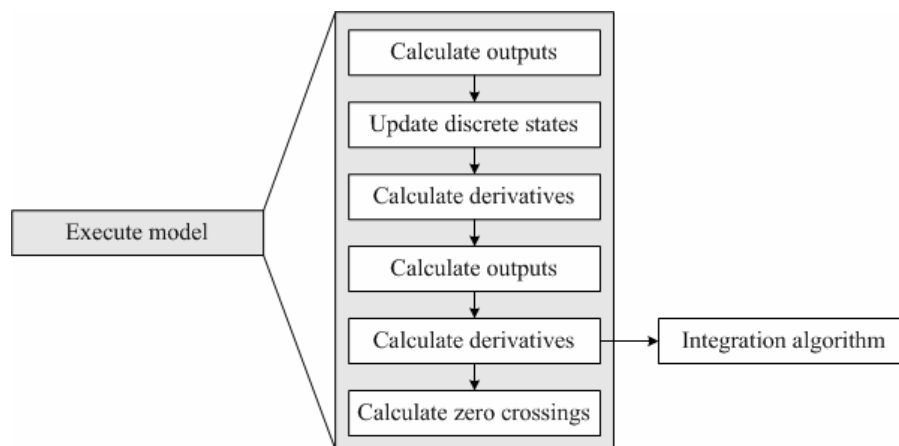
#### **4.4.2.2 *Real-time execution***

Real-Time Workshop takes the Simulink model and generates the application or algorithm code that contains the system of equations derived from the model as well as the block parameters and the code to perform initialization. Real-Time Workshop also provides a run-time interface that allows the model code to be built into a complete, stand-alone program that can be compiled and executed. The model code contains functions that correspond to the applicable simulation steps outlined in Figure 4.7, i.e., compute the model outputs, update the discrete states, integrate the continuous state (if applicable), and update time.

xPC Target generates its own main program. This program interacts with the execution driver for the model code, which in turn calls the functions in model code. The functions then write their calculated data to real-time model. At each sample interval, the main program passes control to the model execution function, which executes one step through the model. This step reads inputs from the external hardware, calculates the model outputs, writes outputs to external hardware, and then updates the states. Figure 4.8 gives the illustration about real-time execution of the model code [112].

If these computations require the plant output of the previous sample time, they must be performed in one sample interval. This implies synchronization between the execution of the logical program by the controller and the dynamic behavior of the plant in real-time. The sample rate is determined by control law analysis and depends on the time constants of the plant. The faster the plant time constants, the higher the required sample rate.

To make the Simulink model designed in simulation step can be used for real-time application, the input and output port have to be appropriately configured for the existing I/O module.



**Figure 4.8:** Real-time execution of the model code.

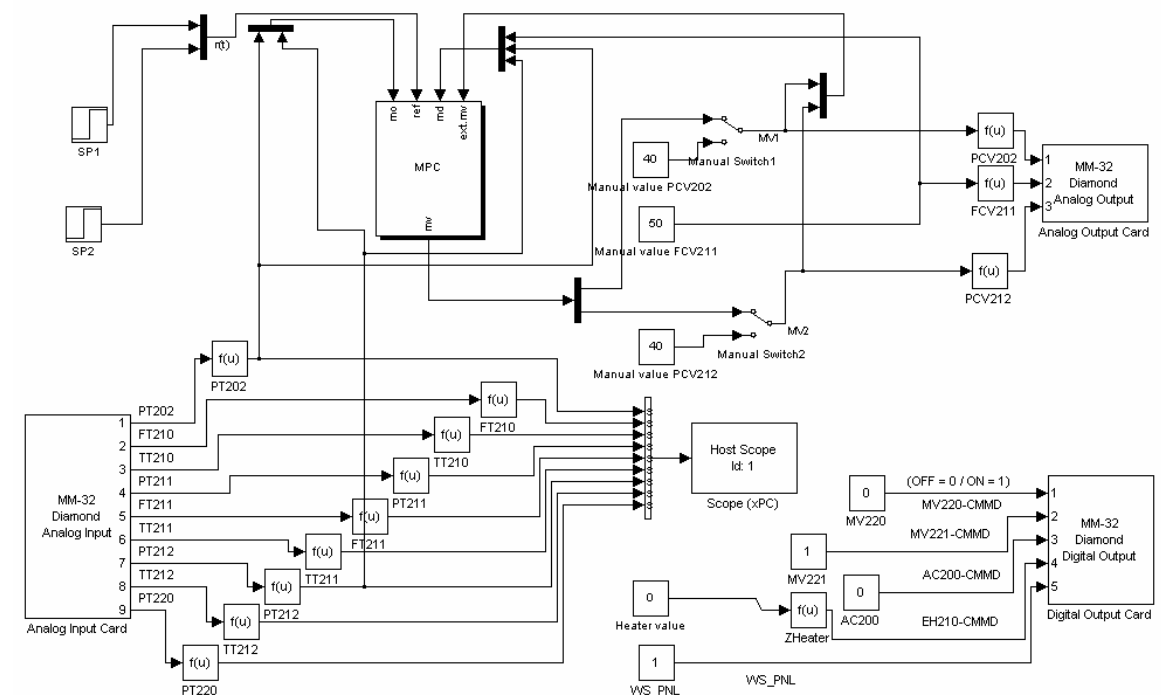
In this work, a Diamond-MM-32-AT analog-to-digital (A/D) and digital-to-analog (D/A) converter is used. This module uses a 16-bit A/D converter. This means that the analog input voltage can be measured to the precision of a 16-bit binary number. Since the maximum value of a 16-bit binary number is  $2^{16} - 1$ , or 65535, so the full range of numerical values that could be obtained from a Diamond-MM-32-AT analog input channel is 0 – 65535. The smallest change in input voltage that can be detected using this device is  $1/2^{16}$  of the full-scale input range. This smallest change results in an increase or decrease of 1 in the A/D code, and so this change is referred to as one least significant bit (1 LSB). Therefore, the resolution of the analog input (1 LSB) for measuring unipolar (positive only) analog voltages that have 0 – 5 Volt full-scale range, as used in this work, is 76  $\mu$ V.

For the analog output, Diamond-MM-32-AT uses a 12-bit D/A converter which able to generate output voltages with precision of a 12-bit binary number. So the full range of



numerical values that can be written to the analog outputs on Diamond-MM-32-AT is 0 to 4095. For a 12-bit D/A converter, the resolution is  $1/2^{12}$  of the full-scale output range. Since the full-scale range of analog output used this work is 0 – 5 Volt, so the resolution of analog output (1 LSB) is 1.22 mV.

The model predictive control algorithm is then prepared to be carried out for the real-time application. Figure 4.9 shows the Simulink diagram which is used to build a model code of MPC controller and then downloaded to xPC TargetBox. As it can be seen, the input of MPC controller is connected to analog input module, and the output of controller is connected to analog output module, which have been configured to receive and transmit 0 – 5 Volt analog signal in every sampling time (1 second).



**Figure 4.9:** Simulink diagram that ready to be downloaded into xPC TargetBox for real-time MPC.

#### 4.4.3 Changing the parameters

While the application is running, xPC Target support the modification of certain parameters in the Simulink blocks, such as set-points values and controller switches. The parameter changes are immediately reflected in the real time application.

Several parameters require re-initialization of the model. In this case, the parameters cannot be changed in real-time mode. The new model code has to be built and the real-time application has to be stopped, so that the xPC Target can replace the existing model code with the new one. Such parameters commonly influences the algorithms, and cannot be categorized as the input or output block of Simulink.

#### **4.4.4 Data logging and online monitoring**

The plant is typically instrumented with sensors, so that the control action can be well calculated by the embedded control system and transmitted to the actuators. However, for several purposes, the data need to be stored in a persistent form or made visible in real time.

xPC Target supports several monitoring and data logging methods, including xPC Target scopes, outport blocks in the Simulink model of target application, and also provide the requested data from MATLAB environment.

xPC Target scopes are data display options, which can be associated with any signal in the target application. In order to be displayed on the monitor, the data that is sent to the scope can be stored in RAM or on the xPC Target file system and transferred to the host when the execution of the target application stops.

Outport blocks in the Simulink model of the target application can be used to log data to an object in the MATLAB workspace once the execution of the target application is terminated. After stored in the workspace, the data can be manipulated as regular workspace variables, saved into a file, and/or displayed in a MATLAB plot. Time and the model state can be logged in the same way in which this data can be written to the MATLAB workspace during operation of the target application.

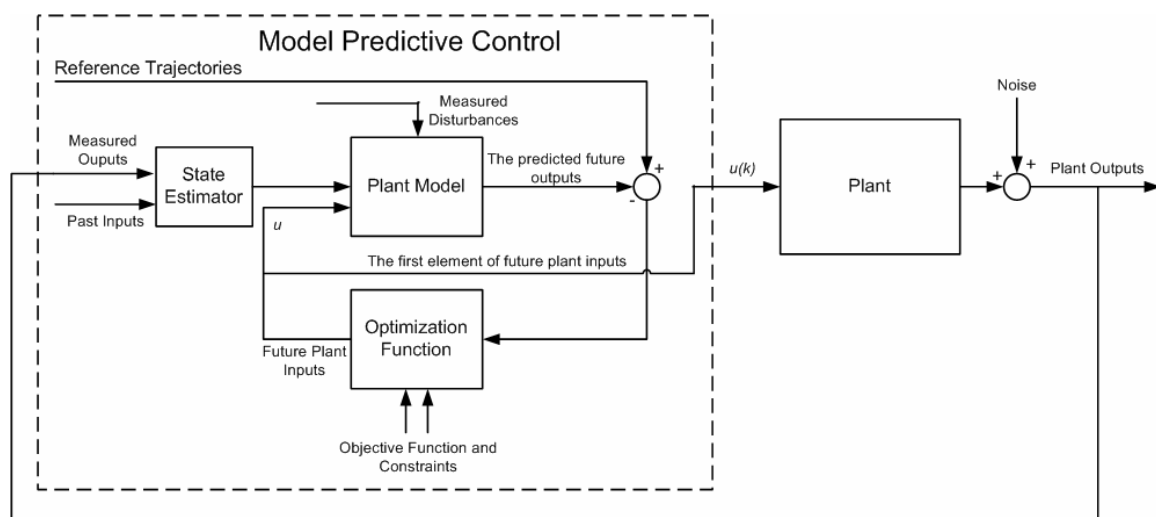
The tight interaction between MATLAB, Simulink, Real-Time Workshop, and xPC Target makes it possible to write a script that monitors the signal outputs. In this work, the script is created to monitors and logs the data in every sampling time (1 second), and then stores the logged data into a spreadsheet file when the application finished. Using this script, the real-time monitoring seems more flexible rather than using xPC Target scopes, since the sampling time can be easily adjusted and the value of signals can be

clearly displayed. Also, when the data need to be examined, it is not necessary to stop the application like when the output block is used as data logger.

## 4.5 Real-Time Implementation of MPC

In previous chapter, the empirical models of gaseous pilot plant have been obtained. The rapid prototyping of an embedded MPC controller has also been presented in previous section. As continuation of the work, the real-time implementation of MPC using the internal models and the experimental environment designed before is presented in this section. This section begins with the analysis of model mismatch, and followed by the real-time tuning of model predictive controller. At the end of this section some discussions are presented.

### 4.5.1 Setpoint tracking: Analysis of MPC under model mismatch



**Figure 4.10:** The schematic diagram of the closed-loop MPC control system.

There are 3 models of gaseous pilot plant have been obtained using subspace identification technique with variation in input signals and model structures, which are named Model 1, Model 2, and Model 3. The first two models have three inputs, i.e., the three control valves, and two outputs, i.e., the upstream and downstream pressures, while Model 3, which is constructed from two MISO models, has five inputs and two outputs. The global comparison of the models has been performed using APRBS response as the validation data, while the best fit criterion is used to measure the performance. From the open-loop validation, it is observed that Model 3 produces the best performance. In this

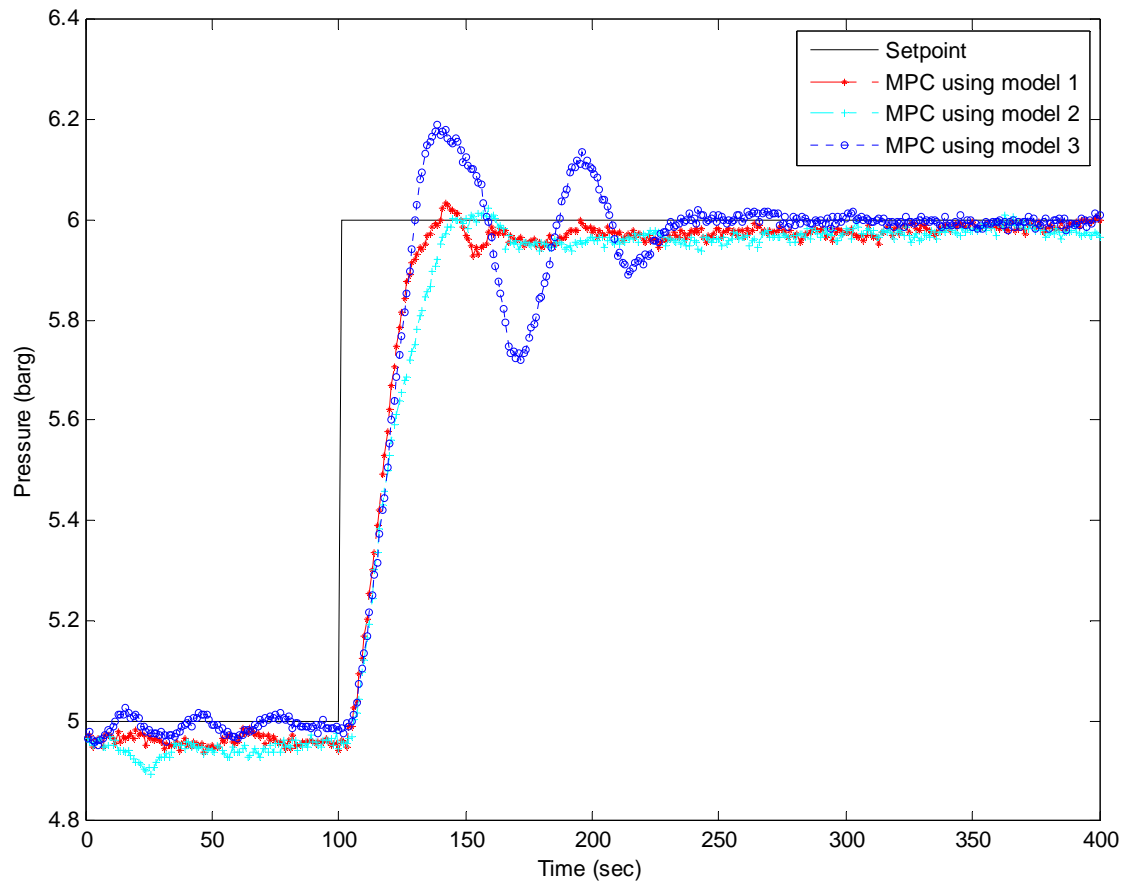
sub-section, those three models are employed as the internal model of MPC controller. The controllers will be tested for setpoints changing in real-time condition, and their performances will be analyzed.

The schematic diagram of the closed-loop system used in this experiment is shown in Figure 4.10. The block of plant model in Figure 4.10 uses the state space plant model as given in (4.5). Using MPC Toolbox, the plant model should be specified in discrete LTI form without including the plant noise and the measurement noise when defining the model predictive controller. If there are any non-manipulated variables contained in the input vector, such variables should be defined as measured disturbances. The state estimator block in the figure represents the observer given in (4.6) to (4.8). In MPC Toolbox, the default parameter for the estimator gain in (4.6) is designed using Kalman filtering techniques by using *kalman* syntax in the MATLAB Control System Toolbox. Using the plant model, the information of current plant inputs, and the estimated state of the plant obtained by the observer, the prediction of plant output for several steps ahead can be calculated using the equations in (4.15) to (4.19). The optimization function block in Figure 4.10 is responsible for solving the control law of MPC problem using QP method. Since MPC Toolbox is used in this research, the constrained MPC problem in (4.56) to (4.60) is solved using QP solver which is available in the toolbox. The MATLAB MPC Toolbox uses Dantzig-Wolfe's algorithm to solve the convex quadratic programming problem. This algorithm can be called using *qp dantz* syntax. The block of plant in Figure 4.10 represents the gaseous pilot plant.

The plant model would be varied from Model 1, Model 2, and then Model 3. All the three controller variations are set to have the same parameters as follows:

- a. Prediction horizon: 40.
- b. Control horizon: 10.
- c. Rate weights ( $\mathbf{R}$ ):  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ; Output weights ( $\mathbf{Q}$ ):  $\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ .
- d. Manipulated variables constraint:  $0 < MV < 100$ .
- e. Process variables constraint:  $0 < PV < 7$  barg.
- f. Sampling time: 1 second.

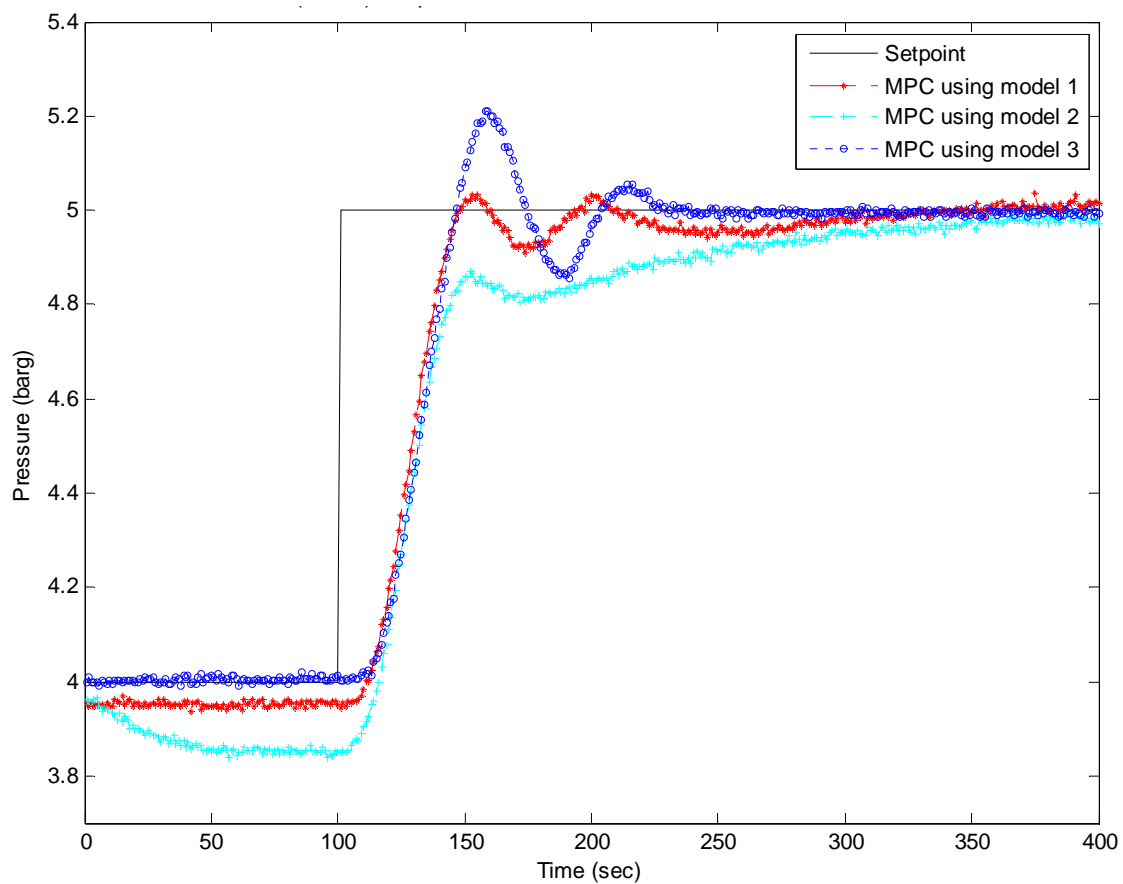
The setting of the controller horizons and the weights of optimization function given above are based on a heuristic approach conducted during this research which implies that such parameters give the best controller responses in the real-time implementation of MPC for controlling the gaseous pilot plant.



**Figure 4.11:** Upstream responses of MPC control system with variation of the internal model.

The controllers are implemented in the plant and given the initial setpoints. The initial setpoint for upstream pressure is 5 barg, while the initial setpoint for downstream pressure is 4 barg. When the process variables are apparently reaching their steady-state conditions, the setpoints are changed. The setpoint of upstream pressure is increased from 5 to 6 barg, while the setpoint of downstream pressure is increased from 4 to 5 barg. The total recorded data is 400 for each process variable, with one second sampling time. The data recording begins from 100 seconds before and 300 seconds after the changing of setpoints.

Figure 4.11 shows the closed-loop responses of upstream pressure (PT202) controlled by MPC using Model 1, Model 2, and Model 3 as the internal model, while for downstream pressure (PT212) the closed-loop responses is shown in Figure 4.12. Several controller performance parameters of this experiment are given in Table 4.1. The settling time parameter used in this experiment is the time at which the controlled variables have entered and remained in the steady-state conditions within 5% error band. The maximum overshoot and steady-state error parameters are presented in scaled value, which are normalized with the difference between the initial and final setpoints.



**Figure 4.12:** Downstream responses of MPC control system with variation of the internal model.

Analyzing the responses of upstream control system given in Figure 4.11 and the controller performance parameters in Table 4.1, the MPC controller using Model 1 and Model 2 give better results in the transient conditions, indicated by zero overshoot and smaller settling time. However, more details in the steady-state conditions, the plant responses of both controllers tend to have offset (steady-state errors). As it can be seen in Table 4.1, MPC using Model 1 produces of 4.5% steady-state error, while MPC using

Model 2, even worse, produces the steady-state error of 6.7%. For MPC using Model 3, an overshoot occurs in the transient condition. The settling time given by MPC using Model 3 in upstream control system is also larger rather than when Model 1 and Model 2 are used. However, the major breakthrough given by Model 3 is that there are no more steady-state errors, neither in initial steady-state nor in the final steady-state conditions.

**Table 4.1:** MPC controllers performance parameters with variation of internal model

The internal model used in MPC	Max. overshoot in scaled value (%)		Settling time (second)		Steady-state error in scaled value (%)	
	PT202	PT212	PT202	PT212	PT202	PT212
Model 1	- *	- *	75	140	4.5	5.7
Model 2	- *	- *	75	200	6.7	15.9
Model 3	28.1	20.5	125	120	0	0

- \*: within 5% error band.

For the response of downstream control system, MPC controllers using Model 1 and Model 2 as the internal model produce the worse response compares to the response of upstream control system, as indicated by greater steady-state errors. Inline with the upstream control system, the steady-state errors, either in the initial or in the final steady-state condition, disappear when Model 3 is used as the internal model of MPC controller. Even better, the settling time of downstream control system using Model 3 as the internal model of MPC is smaller compares to when Model 1 and Model 2 are used. Yet, MPC using Model 3 produces unacceptable overshoot in the transient condition.

Since there are some undesired conditions occurring, such as the overshoot of controlled variables in the transient conditions, it is necessary to fine-tune the MPC controller. The techniques to eliminate the overshoot and fine-tune the MPC controller will be presented in the following sub-section.

#### 4.5.2 MPC parameters tuning

In this sub-section, four tuning parameters will be examined in the real-time implementation of MPC in the gaseous pilot plant. The MPC controller uses Model 3 as the internal model, since it has shown some advantages, especially that it can produce zero steady-state errors for upstream and downstream control system, as described in sub-

section 4.5.1. These tuning parameters are: controller sensitivities, prediction and control horizons, state estimation gain, and reference trajectories.

#### 4.5.2.1 *Controller sensitivities*

The sensitivity of MPC is mainly affected by tuning the weights of optimization function [29]. The effect of these weights can be briefly explained as follow:

- a. Rate weight, or also called penalty on move factor [29]. This weight penalizes changes in the manipulated variables. The plausible values for rate weight are:
  - i. Zero, means that there is no penalty. MV can move anywhere between its upper and lower limits. However, this setting might allow the excessive movements.
  - ii. Positive, means that increasing the weight makes the controller tend to reduce the movement in manipulated variables.
- b. Output weight, or also called penalty on error factor [29]. This weight penalizes deviation of output from its setpoint or reference trajectory. Used as follows:
  - i. Zero, means that there is no penalty. Used when the output does not has a setpoint.
  - ii. Positive, increasing this weight makes the controller tend to keep the output nearer its setpoint.

In this experiment, the diagonal matrices of rate weight and output weight are used:

$\mathbf{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix}$ , and  $\mathbf{R} = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix}$ . The output weight,  $\mathbf{Q}$ , is varied between:  $\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ,

$\mathbf{Q} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$ , and  $\mathbf{Q} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ , while the output weight matrix is kept to be

constant:  $\mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ . So that, it can be said that the approach used in this experiment is

by varying the ratio between  $Q$  and  $R$ .

The other controller parameters in this experiment are set as follows:

- a. Horizons

The short horizons are first considered in this experiment. The prediction horizon of the MPC controller is selected to be 10 and the control horizon is selected to be 5. The effects of longer horizons would be further examined.



b. State estimation gain

The default parameter for the state estimation gain designed using Kalman filtering techniques which is available in MPC Toolbox is used in this experiment. Without specified the noise level of plant, the suggested estimator gain (**K**) for MPC controller using Model 3 is:

$$\mathbf{K} = \begin{bmatrix} 0.0001 & 0.0004 & 0 & 0 & 0.6171 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6180 \end{bmatrix}^T. \quad (4.62)$$

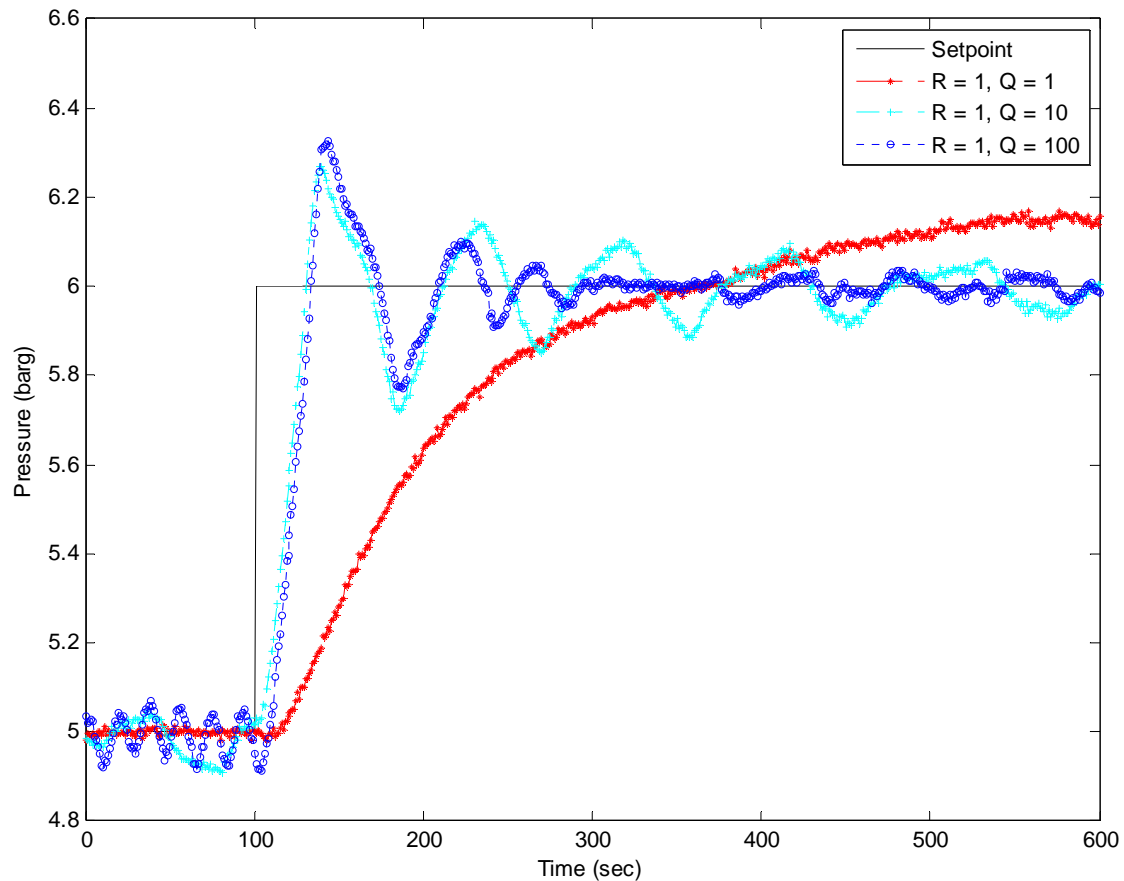
c. Reference trajectories

There are no reference trajectories filter formulation used in this experiment. The MPC controller would be driven by step-shaped setpoints.

Figure 4.13 shows the comparison of upstream responses controlled using MPC with variation in the output weight, while the rate weight is kept to be constant. Several control performance parameters, including condition of process variable in steady-state conditions of upstream control system, are given in Table 4.2. The term ‘NA’ stands for *not available*, which means that the required information cannot be obtained from the logged data, due to the bad performance of control system. For instance, when the controller has the output weight equals to the rate weight, the system is not in steady-state condition yet at 300 seconds after the changing of setpoint. Hence, the settling time of system in this condition cannot be determined. The term ‘SS’ stands for *steady-state conditions*. So that, the term ‘Initial SS’ refers to the condition of process variable in the initial steady-state condition (before the changing of setpoint) and the term ‘Final SS’ refers to the condition of process variable in the final steady-state condition (after the changing of setpoint). As it can be observed, when  $Q$  and  $R$  are set at the same value, which is 1, the response of controller is very slow and has not in the steady-state yet at 300 seconds after the changing of setpoint. When output weight is increased become 10, the response of controller is faster, however, the amplitude of oscillation in the final steady-state condition remains big. The oscillation in the final steady-state condition is much reduced, or even removed, when the output weight is increased become 100.

The comparison of downstream responses controlled using MPC with variation in output weight is illustrated in Figure 4.14, and the controller performance parameters are given in Table 4.3. Inline with the response in upstream, when the rate weight and the output

weight are set to have value of 1, the response is very slow. This implies that the output weight has to be increased. When the output weight is set to be 10, the controller response is much faster, however, there is oscillation remains in the final steady-state condition. A better response is shown when the output weight is increased become 100. The response is fast enough and the oscillation in final steady-state condition is much reduced.



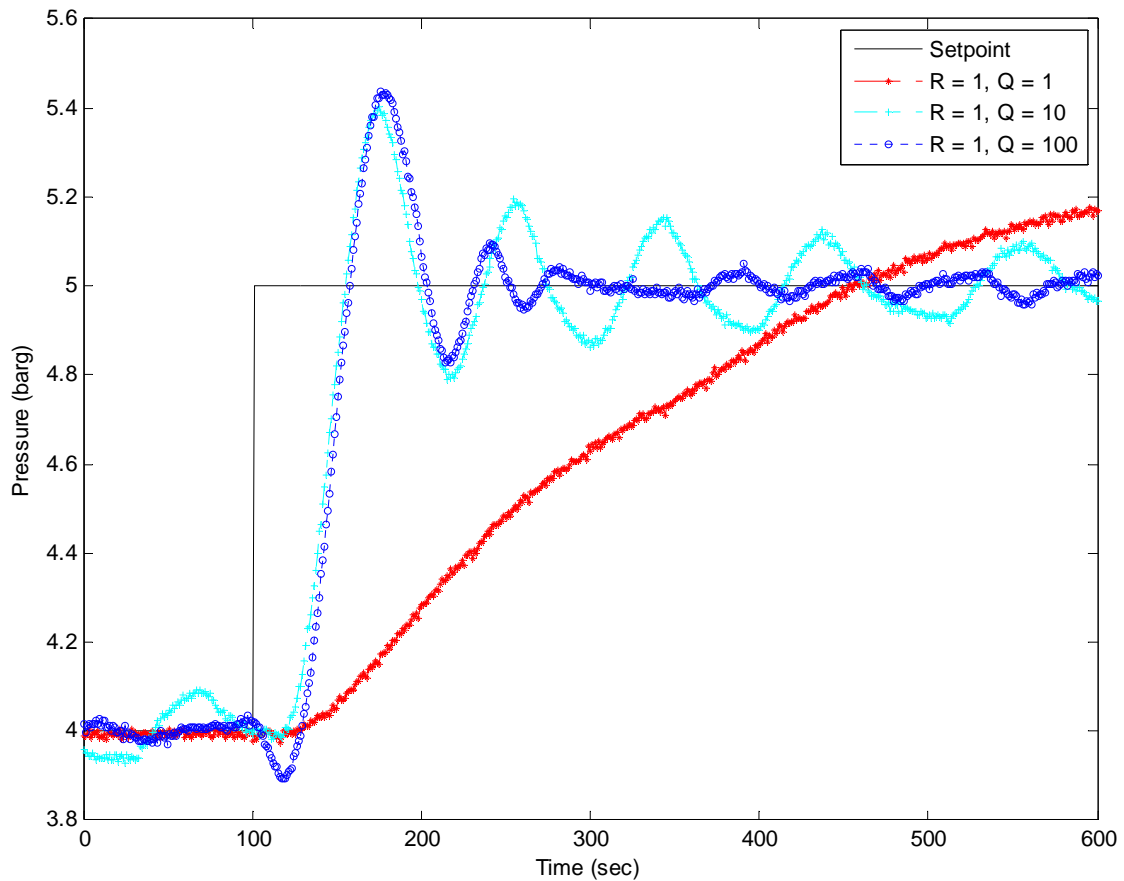
**Figure 4.13:** Upstream responses of MPC control system with variation of the weights of objective function.

**Table 4.2:** MPC controllers performance parameters with variation of the weights of objective function for the upstream responses

The weight of MPC's optimization function	Amplitude of oscillations (%)		Period of oscillations (second)		Max. overshoot (%)	Settling time (second)
	Initial SS	Final SS	Initial SS	Final SS		
$R=1; Q=1$	-*	NA <sup>†</sup>	-*	NA <sup>†</sup>	NA <sup>†</sup>	NA <sup>†</sup>
$R=1; Q=10$	7	10	74.5	95.7	26.1	NA <sup>†</sup>
$R=1; Q=100$	7.1	-*	19	-*	31.9	160

-\* : within 5% error band.

NA<sup>†</sup>: have not entered and remained in steady-state within 5% error band during 500 seconds.



**Figure 4.14:** Downstream responses of MPC control system with variation of the weights of objective function.

**Table 4.3:** MPC controllers performance parameters with variation of the weights of objective function for the downstream responses

The weight of MPC's optimization function	Amplitude of oscillations (%)		Period of oscillations (second)		Max. overshoot (%)	Settling time (second)
	Initial SS	Final SS	Initial SS	Final SS		
$R=1; Q=1$	-*	$NA^{\dagger}$	-*	$NA^{\dagger}$	$NA^{\dagger}$	$NA^{\dagger}$
$R=1; Q=10$	3.8	4.4	100	120	17.9	350
$R=1; Q=100$	-*	-*	-*	-*	20.5	130

-\* : within 5% error band.

$NA^{\dagger}$ : have not entered and remained in steady-state within 5% error band during 500 seconds.

There are also drawbacks of increasing the output weight. As can be observed in Table 4.2 and Table 4.3, increasing the output weight may produce the bigger overshoots. However, in this case, the overshoot increases seem nothing compare to the improvement of control system such as the faster settling time.

Another drawback of increasing the output weight is it may also increase the frequency of oscillation in the initial steady-state conditions. Analyzing the physical aspect of the plant, maintaining the upstream pressure at 5 barg is somehow more difficult rather than at 6 barg, due to the small size of upstream vessel and the fluctuation of air feed.

#### **4.5.2.2 Horizons**

Wojsznis et al. [29] point out that from an implementation stand point, prediction horizon and control horizon are not convenient to be used as tuning parameters. Al-Ghazzawi et al. [27] propose an online tuning method which is found that the proposed method is not to be very sensitive to the closed-loop prediction horizon. However, since both papers present only the simulation works, there is a possibility that perhaps some real-time conditions have not been covered.

The setting of prediction and control horizons contributes a lot in the computational load of controller. From the derivation of MPC algorithm in section 4.3, the size or dimension of the optimization problem that has to be solved by the controller in every sampling time is mainly affected by the length of prediction and control horizons. The computational load of MPC controller can be simply said as the function of the sum total of prediction horizon and control horizon, even though other variables also have some contributions. Therefore, it is necessary to examine the effect of prediction horizon and control horizon in the closed-loop responses of MPC.

In this experiment, three variations of prediction and control horizons in real-time model predictive control will be examined. First, the prediction horizon will be set to have the value of 10, while the control horizon will be set to have the value of 5. In the second variation, the prediction horizon is lengthen become 40, while the control horizon is set to have the value of 10. This setting is rather intuitive since the value of 5 of control horizon seems to be very small when the prediction horizon is 40. The last variation is by setting the control horizon of 20, while the prediction horizon remains 40.

The other controller parameters in this experiment are set as follows:

##### **a. Controller sensitivities**

The rate weight of MPC optimization function in this experiment is chosen to be 1 and its output weight is chosen to be 100. This setting based on the results given in

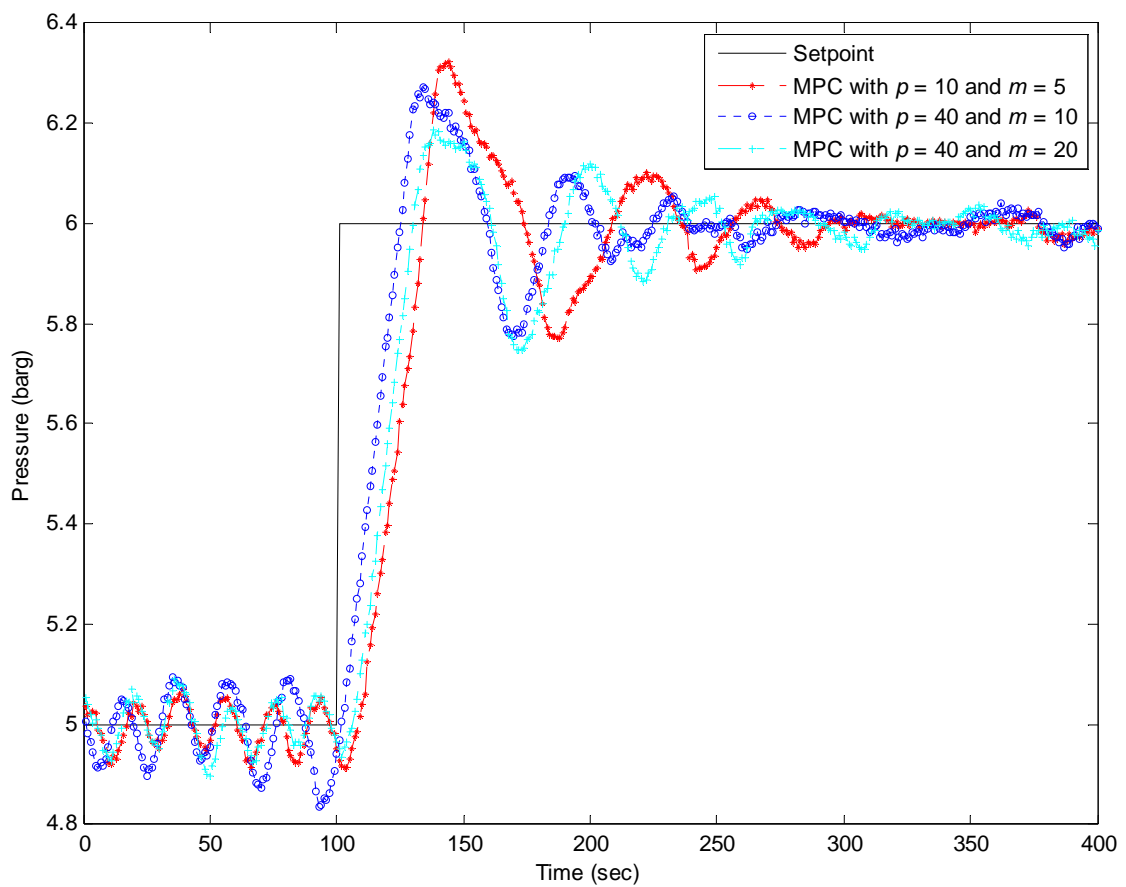
Table 4.2 and Table 4.3, where among the three variations the best controller performances are obtained when  $R$  equals to 1 and  $Q$  equals to 100.

b. State estimation gain

The default parameter for the state estimation gain given in (4.62) remains used.

c. Reference trajectories

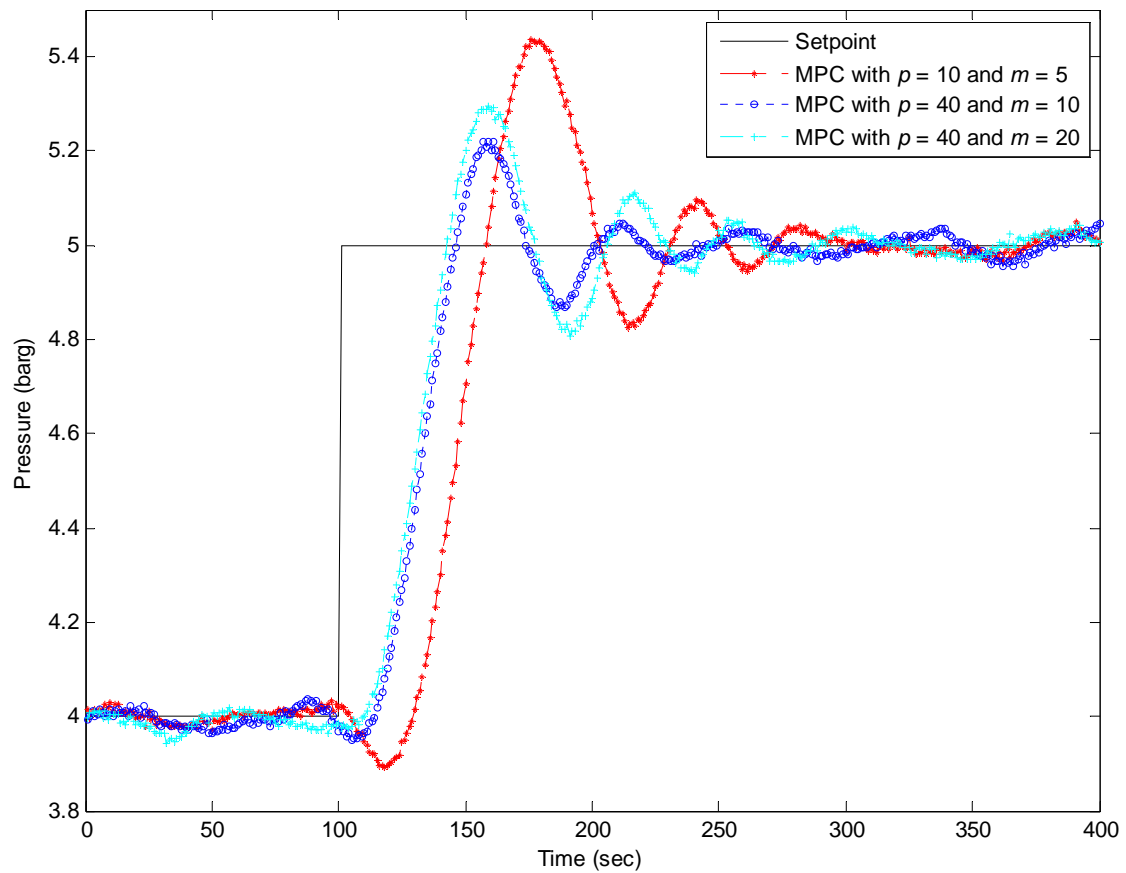
In this experiment, no reference trajectories filter formulation are used. The step-shaped setpoints would be directly given to the MPC controller.



**Figure 4.15:** Upstream responses of MPC control system with variation of prediction and control horizons.

Figure 4.15 shows the comparisons of upstream response controlled using MPC with variation in prediction horizon ( $p$ ) and control horizon ( $m$ ), while for downstream response is shown in Figure 4.16. The controller performance parameters for the upstream

control system using MPC with variation in prediction and control horizons are given in Table 4.4, and for the downstream control system it is given in Table 4.5.



**Figure 4.16:** Downstream responses of MPC control system with variation of prediction and control horizons.

**Table 4.4:** MPC controllers performance parameters with variation of prediction and control horizons for the upstream responses

The horizons of MPC	Amplitude of oscillations (%)		Period of oscillations (second)		Max. overshoot (%)	Settling time (second)
	Initial SS	Final SS	Initial SS	Final SS		
$p = 10; m = 5$	6.3	- *	19	- *	30.5	150
$p = 40; m = 10$	8.4	- *	23	- *	25.3	110
$p = 40; m = 20$	7.9	- *	27	- *	16.8	140

- \*: within 5% error band.

Analyzing the upstream and downstream responses, it can be said that setting a longer prediction horizon improves the performance of control system. A clearer illustration is given by the downstream responses, since the upstream responses are more fluctuating and oscillatory. From the downstream responses, it is observed that by increasing the prediction horizon from 10 to 40, the overshoot can be reduced. When the control horizon is increased from 10 to 20, the bigger overshoot is produced. So that, setting the control horizon closer to prediction horizon may reduce the performance of control system, since the free response of system can be less considered.

**Table 4.5:** MPC controllers performance parameters with variation of prediction and control horizons for the downstream responses

The horizons of MPC	Amplitude of oscillations (%)		Period of oscillations (second)		Max. overshoot (%)	Settling time (second)
	Initial SS	Final SS	Initial SS	Final SS		
$p = 10; m = 5$	-*	-*	-*	-*	42.9	160
$p = 40; m = 10$	-*	-*	-*	-*	20.2	100
$p = 40; m = 20$	-*	-*	-*	-*	28.6	140

-\*: within 5% error band.

During the experiment, it is found that the maximum computational size which can be handled by the prototyping environment in every sampling time is when  $p$  equals to 40 and  $m$  equals to 20. Hence the investigation on the performance of MPC controller with prediction horizon more than 40 steps is not conducted in this research.

#### 4.5.2.3 State estimation gain

Wojsznis et al. [29] also note that model mismatch could produce chattering of the controller output. Significant chattering of the output can be eliminated after the application of model filter with a certain time constant relative to MPC scans.

In this work, due to the noisy environment and the possibility of model mismatch, the chattering and oscillation of controlled output are also the important aspects that need to be considered when fine-tune the MPC controller. However, using state-space MPC, there is a filter embedded inside the controller, which is in the state estimation part. Using Kalman filter as the state estimator, the chattering and oscillation of controlled output due

to possibility of model mismatch and noisy measurement can be reduced by decreasing the Kalman gain in the state estimator.

In the early experiments, the MPC controller uses the default state estimation gain ( $\mathbf{K}$ ) as given in (4.62). This value of  $\mathbf{K}$  is obtained using Kalman filtering techniques based on the assumption that the noise on each measured output, manipulated variables, and measured disturbances are white noise with zero mean and unit covariance. However, this setting produces the sustain oscillation especially in the initial steady-state condition of upstream pressure control. This implies that the controller is too sensitive with the noisy condition and, therefore, the state estimator gain has to be reduced. Figure 4.17 and Figure 4.18 show the comparison of upstream and downstream responses controlled using MPC with state estimator gain equals to  $\mathbf{K}$  in (4.62) and when it is reduced to become:

$$\tilde{\mathbf{K}} = 0.4 \times \mathbf{K} = \begin{bmatrix} 0.00003 & 0.00010 & 0 & 0 & 0.24680 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.24720 \end{bmatrix}^T. \quad (4.63)$$

The reducing factor, 0.4, in (4.63) is heuristically chosen to reduce controller sensitivity to the mismatch of measured and predicted outputs, especially due to measurement noises. There is no optimal design conducted in this work to determine such a reducing factor. The setups of other controller parameters in this experiment are given as follows:

a. Controller sensitivities

Based on the results given in Table 4.2 and Table 4.3, the rate weight of MPC optimization function in this experiment is chosen to be 1 and its output weight is chosen to be 100.

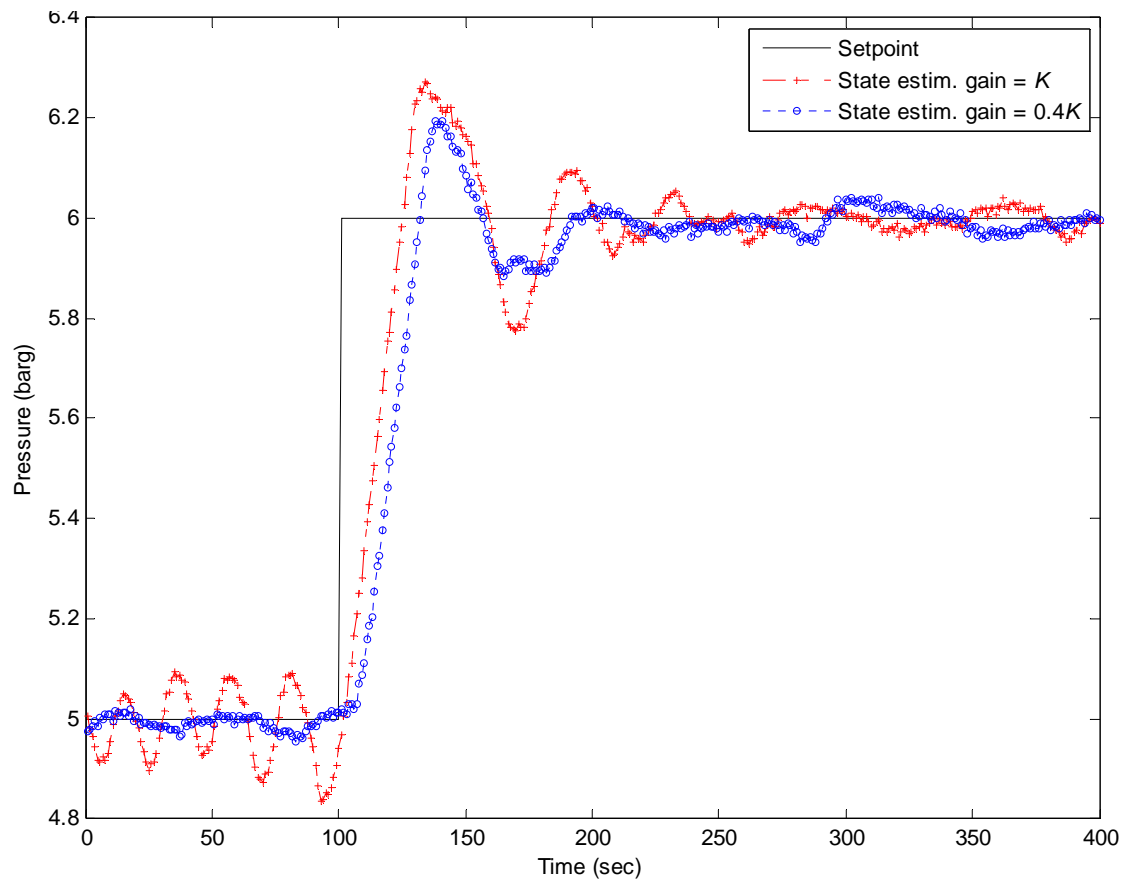
b. Horizons

The settings of controller horizons are based on the results given in Table 4.4 and Table 4.5. Considering the maximum overshoots and settling times of both upstream and downstream responses, the prediction horizon in this experiment is set to be 40 while the control horizon is set to be 10.

c. Reference trajectories

The step-shaped setpoints would be directly given to the MPC controller, hence, no reference trajectories filter formulation are used in this experiment.





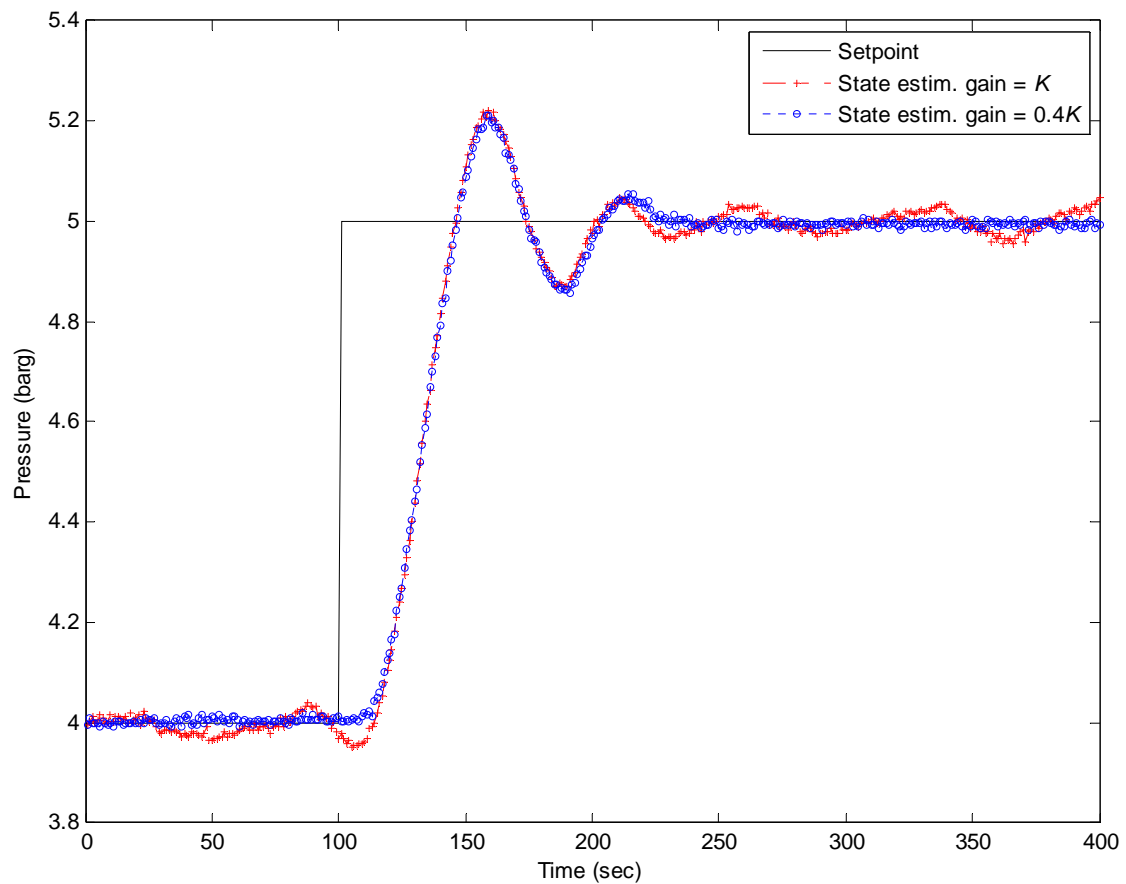
**Figure 4.17:** Upstream responses of MPC control system with variation of the state estimation gain.

The controller performance parameters of MPC using state estimation gain  $\mathbf{K}$  and  $\tilde{\mathbf{K}}$  for the upstream control system are shown in Table 4.6, while for the downstream control system the controller performances are shown in Table 4.7.

**Table 4.6:** MPC controllers performance parameters with variation of the state estimation gain for the upstream responses

The state estimation gain of MPC observer	Amplitude of oscillations (%)		Period of oscillations (second)		Max. overshoot (%)	Settling time (second)
	Initial SS	Final SS	Initial SS	Final SS		
$\mathbf{K}$	9.5	-*	25	-*	23.8	125
$\tilde{\mathbf{K}}$	-*	-*	-*	-*	17	90

-\*: within 5% error band.



**Figure 4.18:** Downstream responses of MPC control system with variation of the state estimation gain.

**Table 4.7:** MPC controllers performance parameters with variation of the state estimation gain for the downstream responses

The state estimation gain of MPC observer	Amplitude of oscillations (%)		Period of oscillations (second)		Max. overshoot (%)	Settling time (second)
	Initial SS	Final SS	Initial SS	Final SS		
<b>K</b>	- *	- *	- *	- *	21.1	100
<b><math>\tilde{K}</math></b>	- *	- *	- *	- *	21.1	100

- \*: within 5% error band.

A clearer illustration of the effect of setting the state estimation gain in MPC is given by upstream control system responses. As can be observed, reducing the state estimator gain removes the sustain oscillation, especially in the initial steady-state condition, and reduces the overshoot in the transient condition. For the downstream responses, however,

the difference of controller performance with the variation of state estimator gain is not significant.

Based on the experiment results, it can be deduced that the state estimator gain plays an important role when the controller has to deal with the noisy environment, while for more stable condition it does not significantly affect the controller performances.

#### **4.5.2.4 Reference trajectories**

Qin and Badgwell [123] point out that almost all of the industrial MPC controllers provide the option to drive the process variable to a fixed setpoint, which makes the objective function penalizes deviations of output on both sides. In practice this type of controller setting is very aggressive and may lead to very large input adjustments, unless the controller is detuned in some manner. This is particularly important when the internal model significantly differs from the process.

Several industrial MPC algorithms provide a process variable reference trajectory option [123]. The reference trajectory idea is to bring the process variable up to the setpoint more slowly, which makes the system able to avoid overshoot. A first order curve is drawn from the current process variable value to the setpoint, with the speed of response is determined by reference trajectory's time constant. Future process variable deviations from the reference trajectory are penalized.

In this experiment, the idea of reference trajectories filter formulation to remove the overshoots in upstream and downstream responses would be examined. During the experiment, other MPC controller parameters are set as follows:

##### **a. Controller sensitivities**

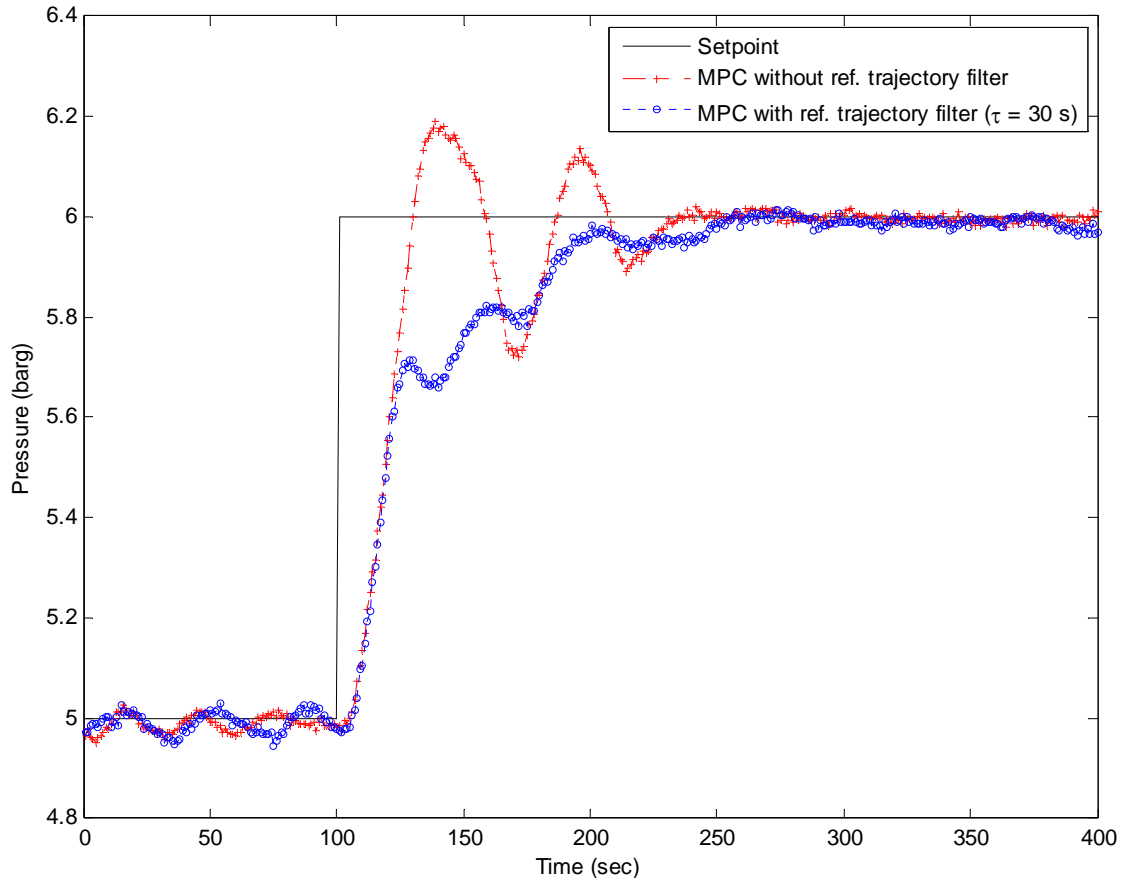
Based on the experiment results in 4.5.2.1, the rate weight and the output weight of MPC optimization function in this experiment is chosen to be 1 and 100 respectively.

##### **b. Horizons**

The prediction horizon in this experiment is set to be 40 while the control horizon is set to be 10. The settings are based on the experimental results in 4.5.2.2.

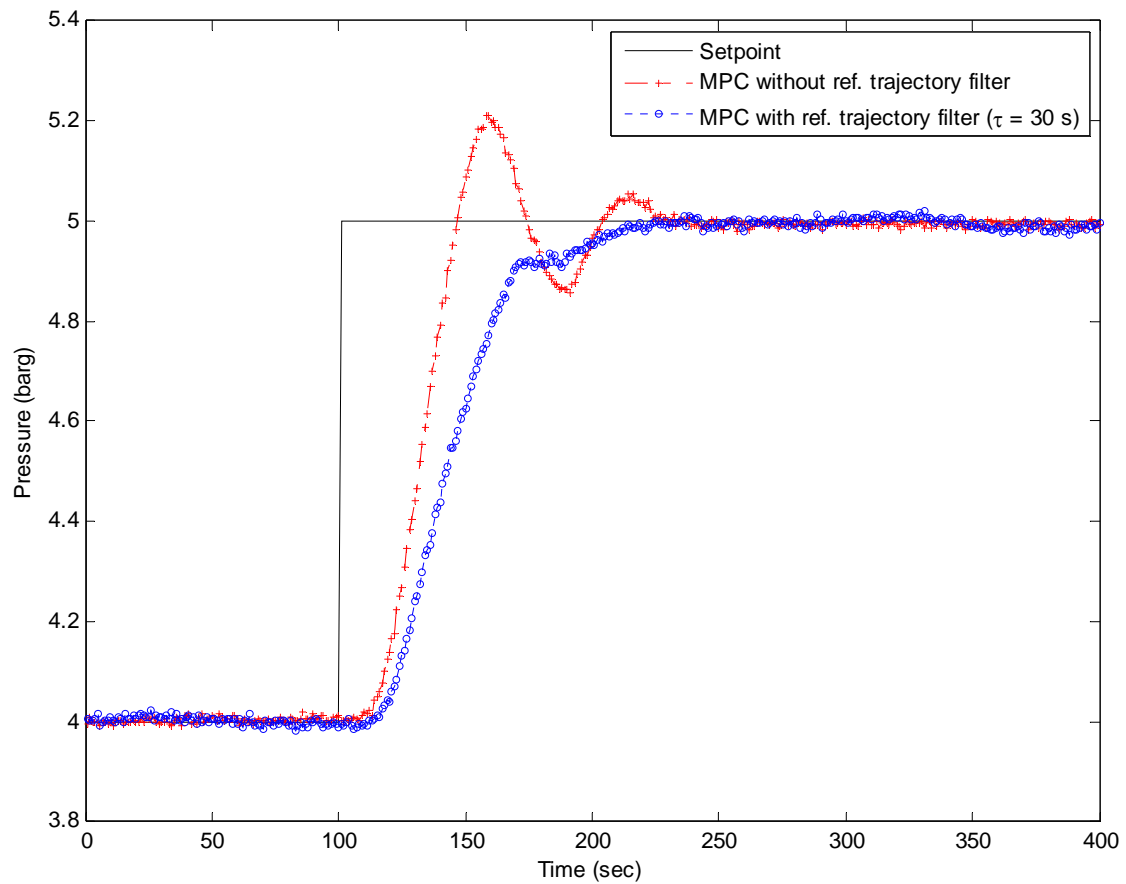
c. State estimation gain

Based on the controller performance parameters given in Table 4.6 and Table 4.7, the state estimation gain in this experiment is set to have the value of  $\tilde{\mathbf{K}}$  as given in (4.63).



**Figure 4.19:** Upstream responses of MPC control system with and without applying the reference trajectory filter.

Figure 4.19 and Figure 4.20 show the comparisons of upstream and downstream responses without and with applying the reference trajectory with the filter time constant  $\tau = 30$  seconds. The controller performance parameters of this comparison are given in Table 4.8. As it can be observed, the overshoots in the transient conditions can be eliminated by applying the reference trajectories filter. More interestingly, in this case, the settling time of upstream and downstream control system are also faster when the reference trajectories filter are applied in MPC.



**Figure 4.20:** Downstream responses of MPC control system with and without applying the reference trajectory filter.

**Table 4.8:** MPC controllers performance parameters with and without applying the reference trajectories filter

Plant outputs	Without ref. trajectory		With ref. trajectory ( $\tau = 30$ s)	
	Overshoot (%)	Settling time (s)	Overshoot (%)	Settling time (s)
Upstream (PT202)	28.1	125	-*	95
Downstream (PT202)	20.5	120	-*	100

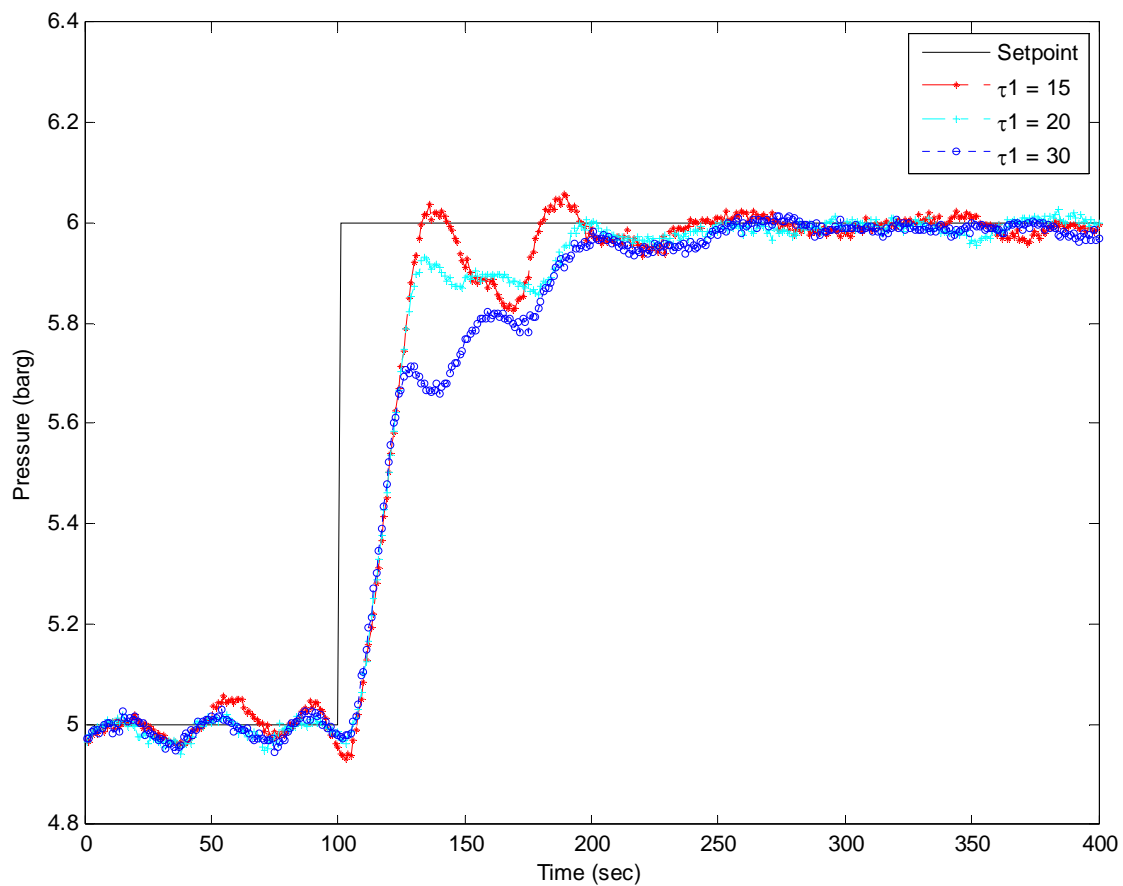
-\* : within 5% error band.

The disadvantage of reference trajectory formulation is that it penalizes the output when it happen to drift too quickly towards the setpoint, as might happen in response to an unmeasured disturbance. So that, it is also important to properly tune the time constant of reference trajectories.

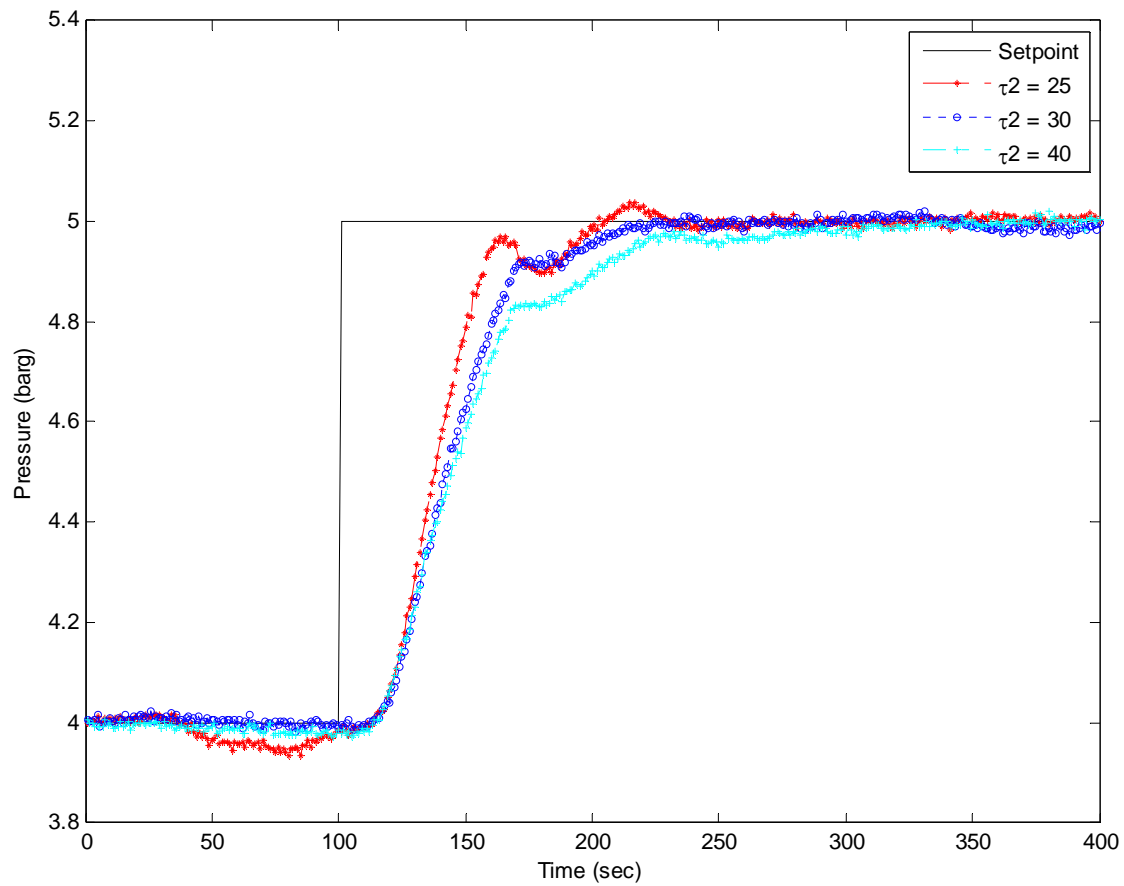
Wojsznis et al. [29] also point out that modifying the setpoint reference trajectory is the primary and intuitive method of online MPC tuning. As the reference trajectory time constant increases, the controller will be able to tolerate larger model mismatch and it also increases overall controller robustness.

In this experiment, three variations of the reference trajectory time constant are examined. For upstream control system, the time constant is varied from 15, 20, to 30 seconds, while for downstream control system, the time constant is varied from 25, 30, to 40 seconds.

Figure 4.21 and Figure 4.22 show the effect of different time constant of reference trajectories in the upstream and downstream responses, and the controller performance parameters of this variation are given in Table 4.9. Based on this experiment results, it can be implied that setting the proper time constant of reference trajectory requires the knowledge about the system, so that the closed-loop response can have the shape as it is desired.



**Figure 4.21:** Upstream responses of MPC control system with variation of the reference trajectory filter.



**Figure 4.22:** Downstream responses of MPC control system with variation of the reference trajectory filter.

**Table 4.9:** MPC controllers performance parameters with different time constant of the reference trajectories

Performance parameters	Upstream pressure (PT202)			Downstream pressure (PT212)		
	$\tau = 15$ s	$\tau = 20$ s	$\tau = 30$ s	$\tau = 25$ s	$\tau = 30$ s	$\tau = 40$ s
Overshoot (%)	5	-*	-*	3	-*	-*
Settling time (s)	70	90	95	95	100	120

-\*: within 5% error band.

### 4.5.3 Discussion

Following the development and open-loop validation of the models, which has been presented in Chapter 4, the implementation of a linear MPC using the developed state-space models to examine their performances in a real-time closed-loop control system is presented in the first part of this section. As the main result, MPC controller using

Model 3 is able to produce zero steady-state errors. The steady-state errors happen since there are some differences between the prediction and the actual plant outputs. These differences are caused by the presence of nonlinearities in the plant that cannot be covered by the internal model of controller. When Model 3 is used as the internal model of MPC, the steady-state errors either in upstream or downstream responses disappear. This implies that during the steady-state conditions, the prediction of plant outputs produced by Model 3 is equal with the actual outputs of plant. It proves that taking into account of the measurement of the upstream output as one of the inputs to predict the downstream output, and vice versa, gives the effective prediction and control over an interacting series process, in this case the gaseous pilot plant.

In the practical process control, zero steady-state error commonly is one of the necessary conditions for control system design. The results show that Model 3 is robust against the nonlinearities in the gaseous pilot plant, indicated by zero steady-state errors during the real-time implementation of a linear model predictive control. Based on the results, it can be concluded that among the three approaches, the technique used to construct Model 3 is the most suitable approach for producing the linear model of an interacting series process which is robust against the plant nonlinearities.

In the second part of this section, four parameters have been examined to illustrate the effect of tuning these parameters to the closed-loop responses of MPC in real-time conditions. The tuning study is started by examining the controller sensitivities. For a linear MPC with diagonal matrices of weights, the controller sensitivities are mainly affected by the ratio of output weight to rate weight. Increasing the ratio of output weight to rate weight increases the controller sensitivity, and on the other hand, decreasing the ratio of output weight to rate weight decreases the controller sensitivity.

The second step of tuning the MPC controller is on the effect of the MPC's horizons. As the objective of MPC is to minimize the error over a finite-time prediction based on the predicted output and produce a set of optimal control action. Therefore, a different setting of prediction horizon and control horizon can results a different control action. Based on the experiment results, the longer prediction horizon produces the better control action, since the controller will consider a longer future behavior of process output. The control horizon has to be less or equal than the prediction horizon. However, a closer control horizon to the prediction horizon reduces the controller computation to consider the free



response of the system, and it can reduce the performance of control system. Setting a very short control horizon may also cause the controller output less optimal.

Both the prediction horizon and control horizon have a big contribution in the computational load of controller, hence it is also necessary to allocate the task of controller computation. It is suggested to set a long enough prediction horizon, so that further increment has no significant effect on the control performance, and control horizon is somehow chosen so that the free response of the system can be well considered.

The third tuning parameter examined in this work is state estimation gain. The necessity to investigate such a tuning parameter arise due to the presence of significant chattering or rapid oscillations in the upstream response when the plant is controlled using MPC, which does not occur in the downstream response. Recalling some information from the previous chapter, the signal-to-noise ratio of upstream pressure measurement is about a half of the signal-to-noise ratio of downstream pressure measurement. Furthermore, in the open-loop model validation, the best fit criterion for upstream data is slightly lower rather than for the downstream data. These conditions can be considered as the cause of the chattering and oscillations in the upstream response. However, it implies that the controller is too sensitive with such noisy condition and model error. Reducing the state estimation gain of the controller proves to solve such a problem. From another investigation of plant responses, it is also found that such an approach gives no significant effect on the controller performance for more stable and less noisy system, in this case the downstream response.

Tuning the horizons, the controller sensitivity, and the state estimation gain of MPC controller give significant improvements to the controller performances in the transient as well as the steady-state conditions. However, the tolerable overshoots remain occur in the upstream and downstream responses when the setpoints are changed. To eliminate the overshoots, the adjustment of the reference trajectories is investigated, resulting that the implementation of the proper reference trajectories can eliminate such an overshoot. The necessity to properly tune the time constants of the reference trajectories is also addressed in this work, since the reference trajectory formulation penalizes the output when it happen to drift too quickly towards the setpoint, which may makes the responses have the undesired shapes.

Based on the results obtained from varying the MPC tuning parameters, the important features to be considered when tuning a MPC controller have been presented in this work. A set of guidelines for tuning an MPC controller as a result during this study is proposed as follows:

- a. If the responses of MPC controller are very slow, or the oscillation of process variables in steady-state conditions are presences with high amplitudes and low frequencies, it is necessary to increase the controller sensitivities by increasing the output weight, or in other hand, reducing the input weight of optimization function.
- b. It is necessary to set a long enough prediction horizon, so that further increment has no significant effect on control performance. The control horizon is someway chosen so that the free response of the system can be well considered. The computational capability of the controller in every sampling time has also to be considered when determining the prediction and control horizons.
- c. Should there a rapid oscillation or chattering of controller outputs in steady-state conditions, it is necessary to reduce the gain of state estimator.
- d. If the undesired overshoots remain occur, the reference trajectories adjustment are worthwhile to be considered.

#### 4.6 MPC and PID

Since the use of PID control in 1910, which extensively addressed in Elmer Sperry's ship autopilot, the popularity of PID control has grown tremendously [101, 102]. The simplicity, clear functionality, and applicability of PID control system have brought it to the top rank of industrial control system algorithm. Nowadays, more than 90% of industrial control systems are still implemented based around PID algorithms [101, 102, 124], and interestingly, most controllers only use the proportional and integral actions [102].

Despite its popularity, a number of limitations in PID control system remain. With advances in digital and computational technology, especially in the industrial applications, the science of automatic control offers varieties of control schemes, such as the kind of centralized optimal control system, which is largely addressed in the application of MPC. Based on its structure and computation algorithm, MPC offers a

number of advantages that no more can be simply handled by PID, such as taking into account of multivariable interactions, actuator limitations, and process constraints explicitly. However, the decentralized control structure of PID and the simplicity of its algorithm yet present some advantages, so that it is remain preferred for certain cases [101-105].

An alternative approach is to combine PID and MPC in the control system. The problem of combining two controllers into a single control system may arise when the controller has to be switched from one to another. Without proper design, sudden movement of manipulated variables with big amplitude commonly will occur. For PID control, many ways have been addressed to overcome this problem, commonly known as anti-windup or bumpless transfer strategy [125-129]. For model-based control system, the bumpless transfer method is commonly addressed based on the type of controller and how the control actions are derived [130-132].

This section presents the study of combining PID and MPC in the control system. The study begins with the tuning of PID controllers parameters using one of the most popular PID tuning methods: Ziegler-Nichols closed-loop tuning method. The controller performances comparison is further performed by examining the process variables and the manipulated variables in the transient and steady-state conditions. Following the work, a simple method to make a bumpless transfer in MPC is demonstrated. In the last part of this section, some discussions are presented.

#### **4.6.1 Tuning of PID**

There are many available methods that can be used to determine the tuning constant values of PID controller. In this work, the Ziegler-Nichols closed-loop tuning method is used. Using such a method it is necessary to find the ultimate gain,  $K_u$ , which is the value of gain at which the loop is at the limit of stability with a proportional-only feedback controller [133]. The period of resulting oscillation is called the ultimate period,  $P_u$ . The controller parameters are then calculated as the function of  $K_u$  and  $P_u$  based on the formulas given in Table 4.10 for the three types of controllers [40, 133].

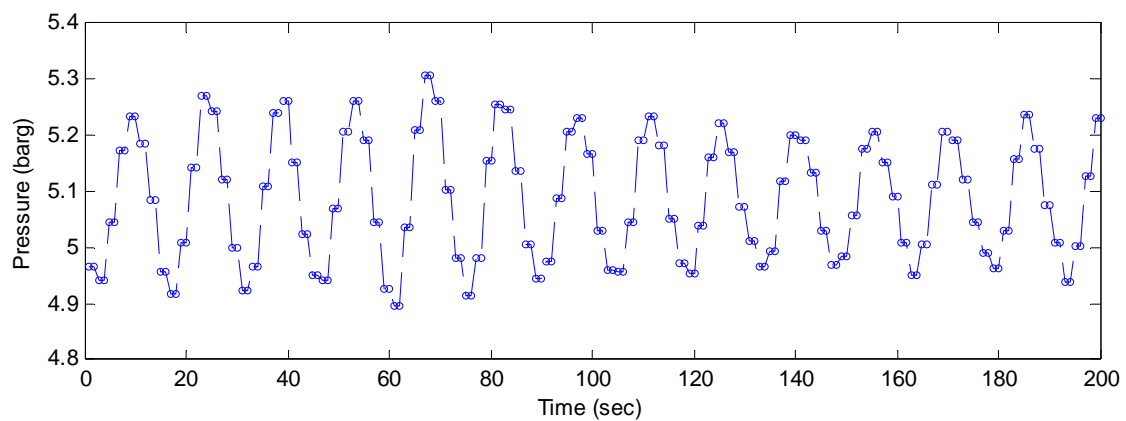
The digital PID controller used in this work has one second sampling time, which is equal to the sampling time used in MPC controller. There are two controllers used, one is for controlling the pressure in upstream, another one is for downstream pressure. The

manipulated variable for upstream pressure is assigned to the inlet control valve (PCV202), while the outlet control valve (PCV212) is selected as the actuator for downstream control system.

**Table 4.10:** Ziegler-Nichols closed-loop tuning correlations

Controller parameters	P-only	PI	PID
$K_c$	$K_u / 2$	$K_u / 2.2$	$K_u / 1.7$
$T_I$	$\infty$	$P_u / 1.2$	$P_u / 2.0$
$T_d$	0	0	$P_u / 8$

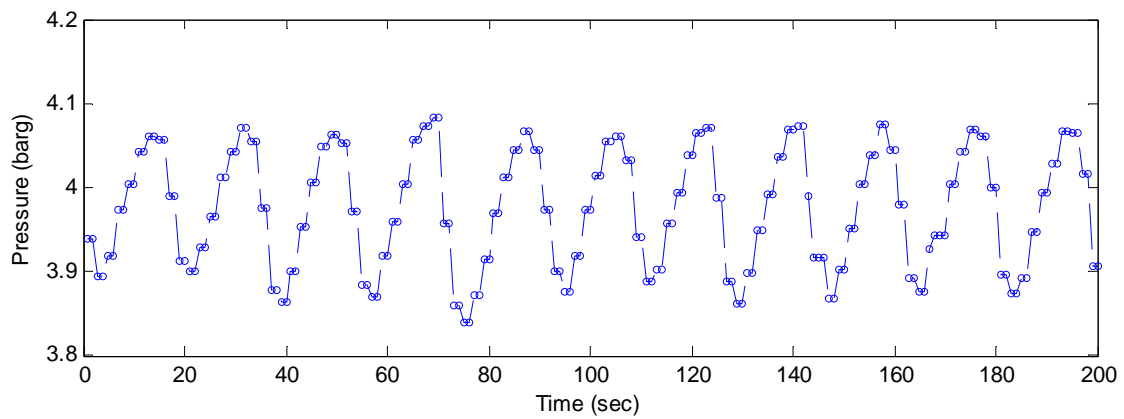
Considering that the process is a kind of fast process, the type of controllers is chosen as proportional and integral only (PI). The derivative mode is selected to be zero, since it provides rapid correction based on the rate of change of the controlled variable, and can cause undesirable high-frequency variation in the manipulated variable [40].



**Figure 4.23:** Sustain oscillation of the upstream process variable (PT202).

The tuning of controllers is done sequentially. The first step is finding  $K_c$  and  $T_I$  parameters for upstream control system. The upstream control system (PIC202) is set in closed-loop proportional-only feedback control, while the downstream control system (PIC212) is in manual mode with constant opening of control valves. From this experiment, the value of controller gain at which the loop is at the limit of stability,  $K_u$ , is 20. Figure 4.23 shows the sustain oscillation of process variable at this condition. The

period of oscillation,  $P_u$ , is 15 seconds. Based on these values,  $K_c$  and  $T_I$  parameters for upstream control system are then determined.



**Figure 4.24:** Sustain oscillation of the downstream process variable (PT212).

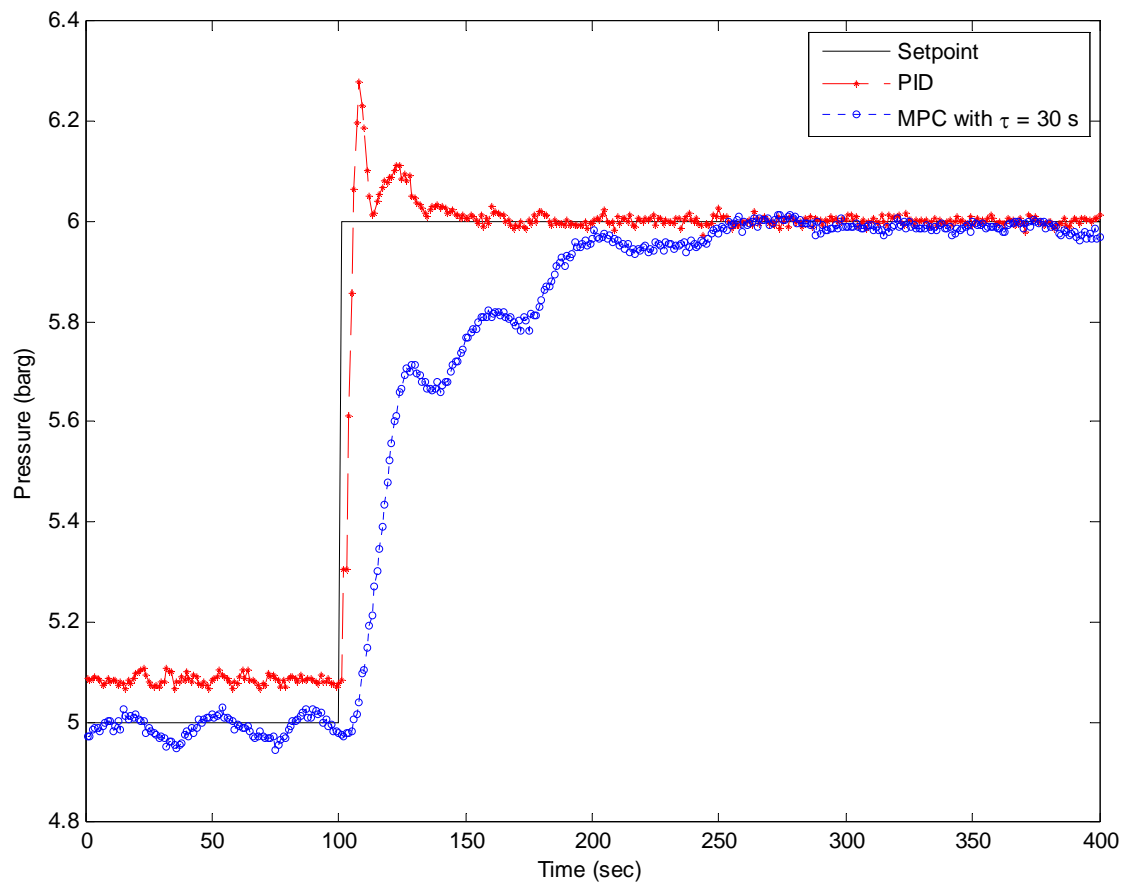
The second step is finding the PI parameters for downstream pressure. The downstream control system (PIC212) is set in closed-loop proportional-only feedback control, while the upstream control system (PIC202) is in manual mode with constant opening of control valves. From the experiment, the controller is at the limit of stability when its proportional gain is 50, and producing the sustain oscillation of process variable with the period is 14 seconds. This sustain oscillation of process variable is shown in Figure 4.24. The parameters of  $K_c$  and  $T_I$  downstream control system are then calculated based on these values. Table 4.11 presents the results of obtaining PID parameters using the Ziegler-Nichols closed-loop tuning method which further be used in this experiment.

**Table 4.11:** Controllers tuning parameters based on the Ziegler-Nichols closed-loop tuning method

Controlled variables	Sustain oscillations		Controllers parameters		
	$K_u$	$P_u$	$K_c$	$T_I$	$T_d$
Upstream pressure (PT202)	20	15 s	9	12.5 s	0
Downstream pressure (PT202)	50	14 s	22.7	11.7 s	0

#### 4.6.2 Control performances

The PID controller's tuning parameters obtained using Ziegler-Nichols method are then implemented in the gaseous pilot plant, and the results are compared with the performance given by MPC controller. Figure 4.25 shows the plant responses in the upstream when the plant is controlled using PID and MPC controllers, while for downstream, the responses are presented in Figure 4.26. The controller performances parameters of PID and MPC in these conditions are given in Table 4.12.

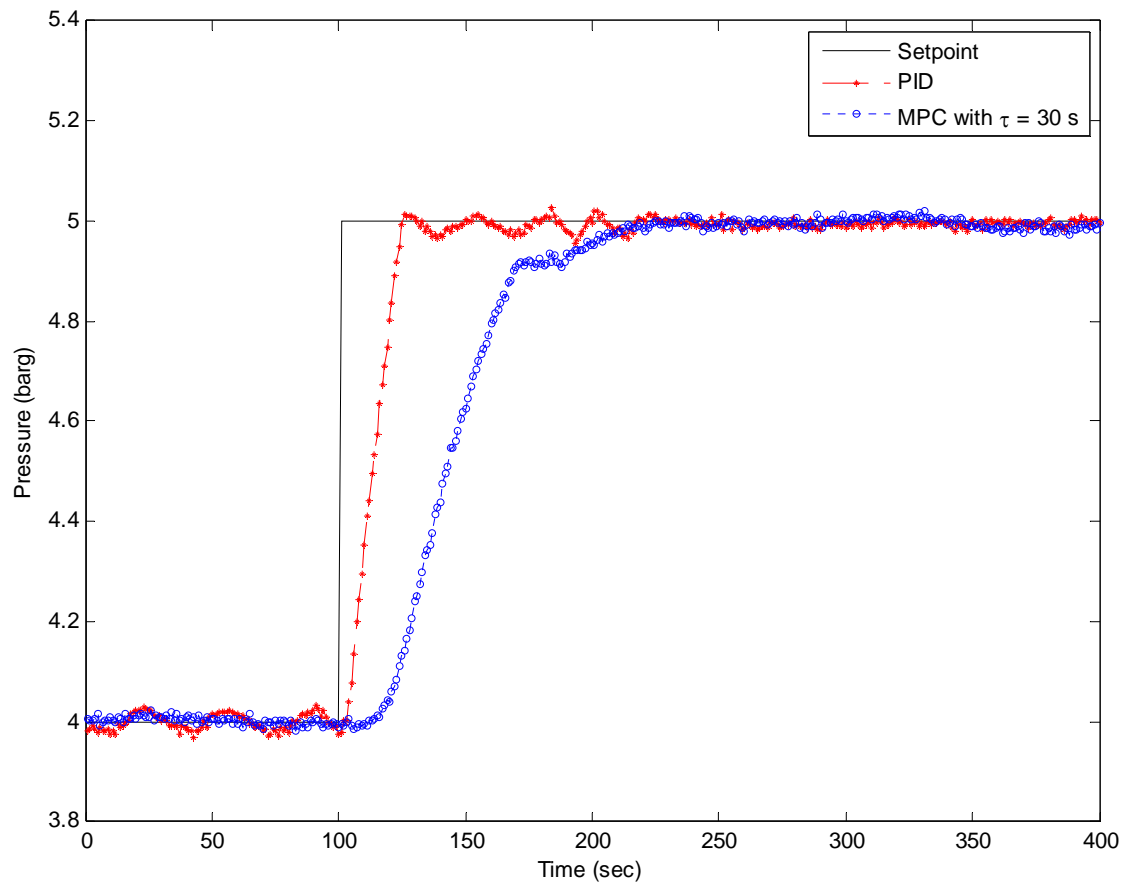


**Figure 4.25:** The comparison of MPC and PID in the upstream control system as the response of setpoint change.

**Table 4.12:** Controllers performance parameters of PID and MPC controllers

The type of controllers	Max. overshoot in scaled value (%)		Settling time (second)		Steady-state error in scaled value (%)	
	PT202	PT212	PT202	PT212	PT202	PT212
PID	26.7	-*	40	25	8.9	0
MPC	-*	-*	95	100	0	0

-\*: within 5% error band.



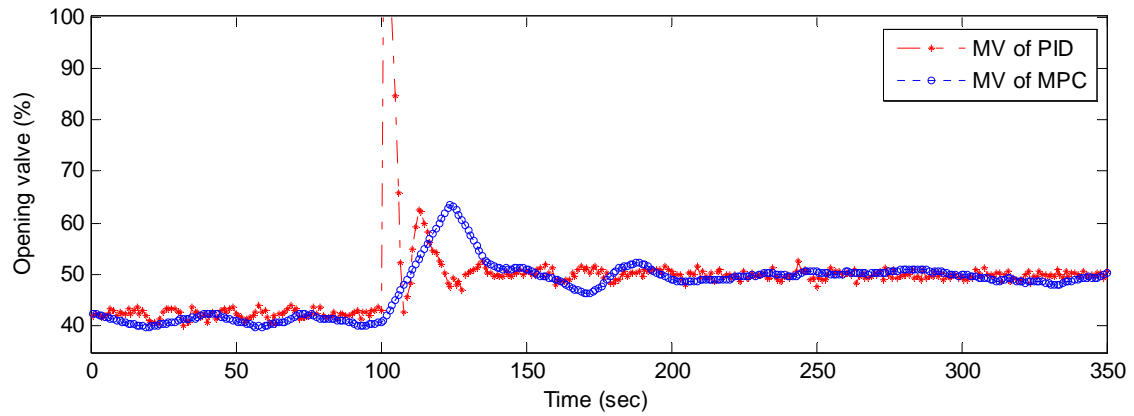
**Figure 4.26:** The comparison of MPC and PID in the downstream control system as their responses to setpoint change.

Analyzing the response of upstream pressure, the PID controller responds rapidly to the changing in setpoint. The overshoot occurs in the transient response, but the controller quickly bring the process variable back to the setpoint. However, there is a steady-state error occur in the first initial condition. Using MPC controller, there is no steady-state errors in the initial and final steady-state conditions, however, the response of the controller is slightly slower.

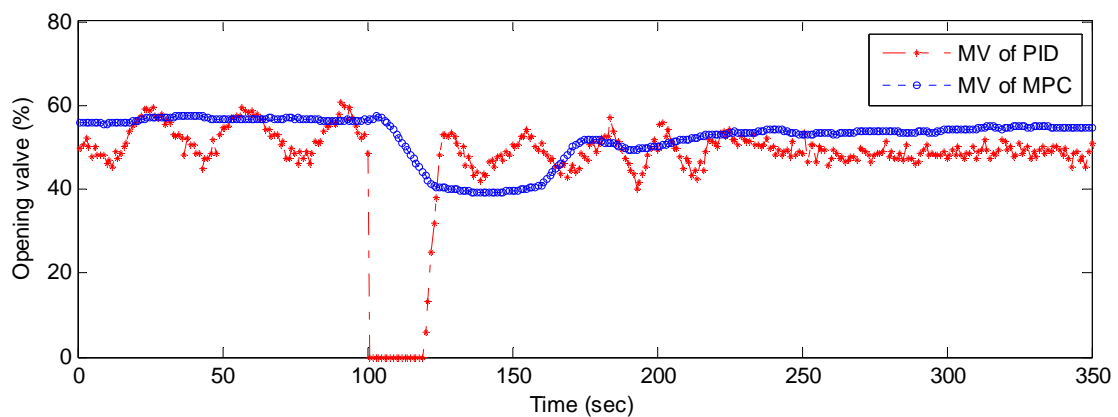
In the downstream, the PID controller shows better response compares to the upstream. The steady-state errors and the overshoot are eliminated. Like in the upstream, the MPC controller is showing slower response compares to the PID controller.

Besides analyzing the responses of process variables, the movements of manipulated variables are also examined. The movement of manipulated variable assigned to the inlet control valve is shown in Figure 4.27, while the manipulated variable assigned to the

outlet control is shown in Figure 4.28. The setpoints of the controllers is changed when the time is equal to 100 seconds. As can be seen, the PID controllers produce fast actions to eliminate these errors, while MPC seems to prevent the rapid movements of manipulated variables.



**Figure 4.27:** The comparison of manipulated variable movements in the upstream control system using PID and MPC.

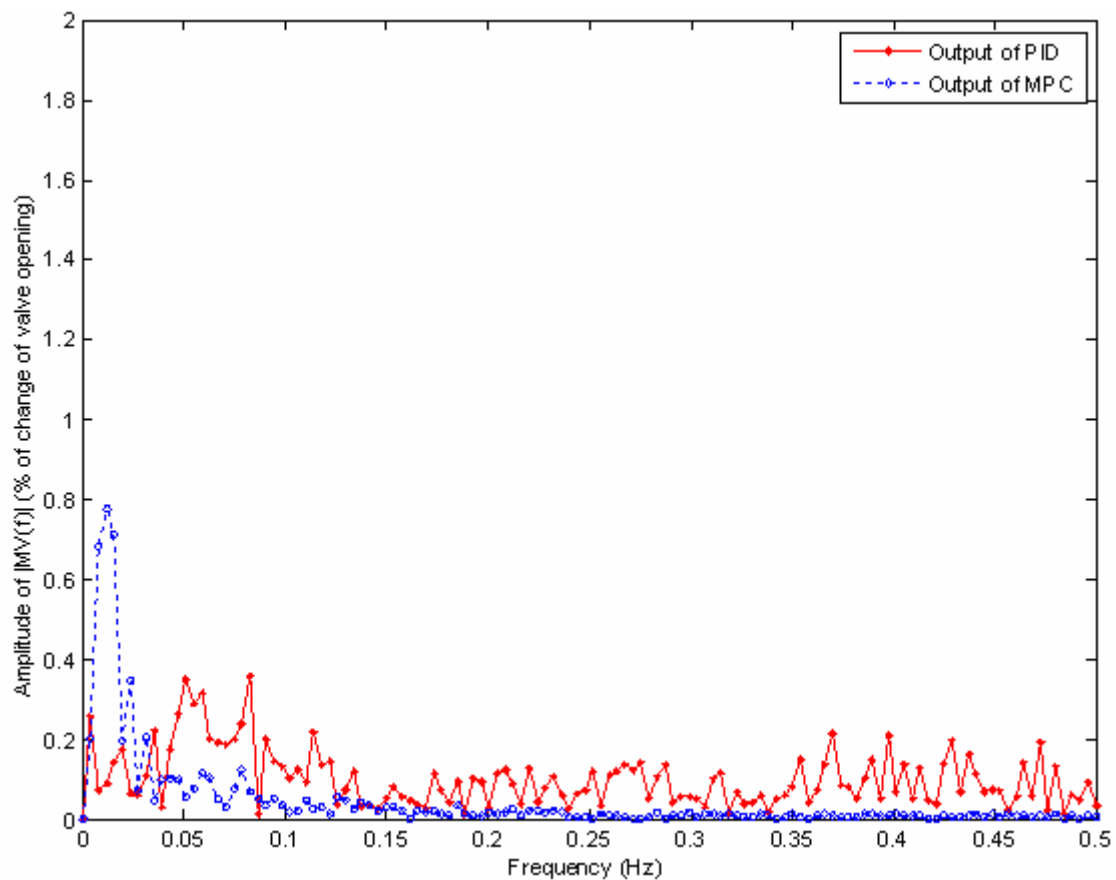


**Figure 4.28:** The comparison of manipulated variable movements in the upstream control system using PID and MPC.

The manipulated variables in the steady-state conditions given by PID and MPC controllers are also examined. For further analysis, these signals are presented in the frequency domain representation. Figure 4.29 presents the spectrums of manipulated variable for the upstream control system in the steady-state condition given by PID and MPC controllers. The signals are taken into account after the settling time, or when the process variable has reached the steady state condition, and normalized to the mean of their steady state value. As can be seen from the spectrum plot, the oscillation of manipulated variable given by PID controller remains presence in high frequencies. In



contrary, the oscillation of manipulated variable given by MPC controller exists only for the low frequencies.

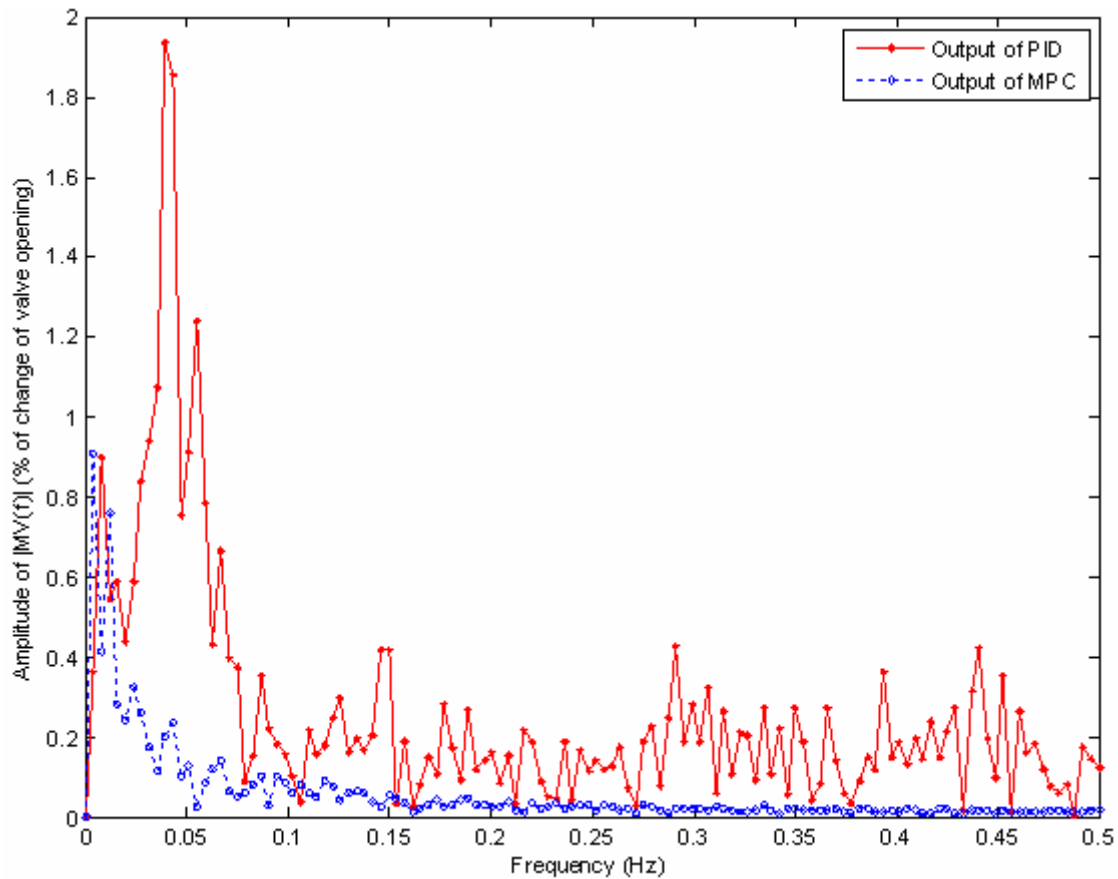


**Figure 4.29:** The spectrum of manipulated variable in steady-state condition given by PID and MPC controllers for the upstream control system.

For the downstream control system, the spectrums of the manipulated variable given by PID and MPC controllers are shown in Figure 4.30. Inline with the spectrums plots of the upstream control system, the movements of manipulated variables produced by PID controllers are still presence in the high frequencies. Furthermore, the oscillation of the manipulated variable given by PID controller indicates the big amplitude. Yet, when MPC controller is used, the movements of manipulated variable are only presence in the very low frequencies.

Since it is not plausible to obtain exactly the true model of the plant which able to capture the whole plant dynamics including plant noises and disturbances, also due to the noisy measurements of the system being controlled, the oscillations of the manipulated variables in the low frequencies cannot be avoided as the controller has to keep the

process variables remain in their setpoints. However, since the manipulated variables will be assigned to the control valves as the actuators, the rapid movements and the high frequency variations commonly suppose to be minimized. Because of their continuous motion, actuators usually undergo wear and aging.



**Figure 4.30:** The spectrum of manipulated variable in steady-state condition given by PID and MPC controllers for the downstream control system.

There are several ways to improve the performance of the PID controller and many guidelines are available [40], however, it will not be investigated in this research. Nevertheless, a fair benchmark or comparison between two controllers requires a deep study of both controllers and plant's behaviors.

#### 4.6.3 Switching control and bumpless transfer

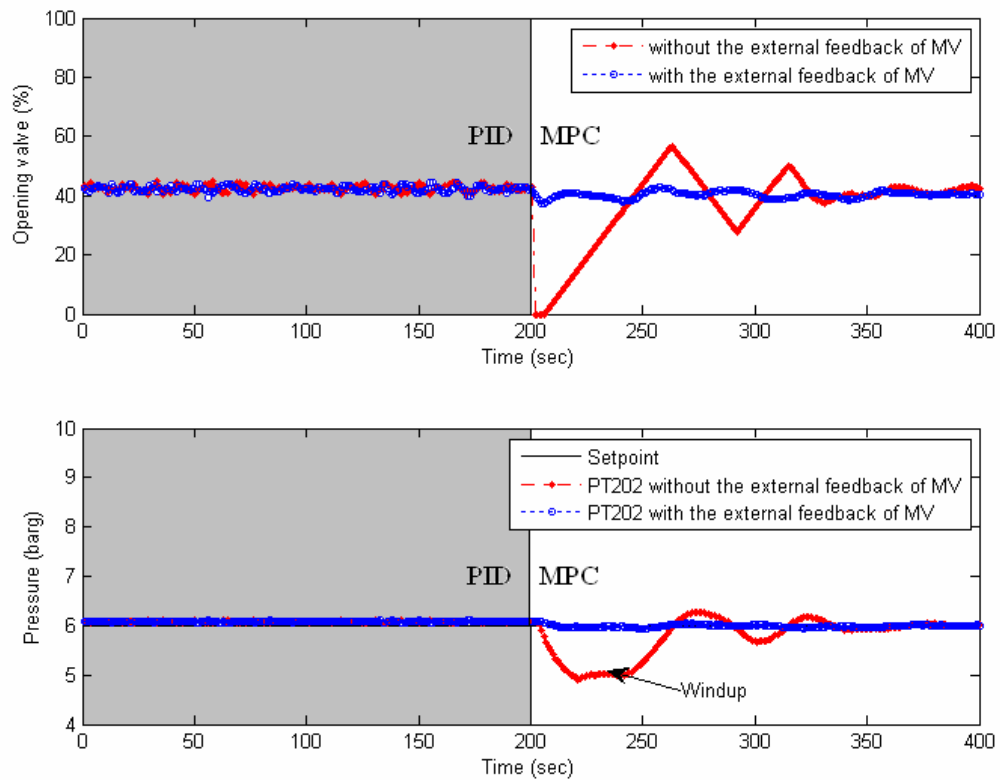
From results of the experiment, it is noted that both PID and MPC has their own advantages. It depends on the users to select the controller based on the kind of performance to achieve. However, there is also an option to use both controllers, PID and



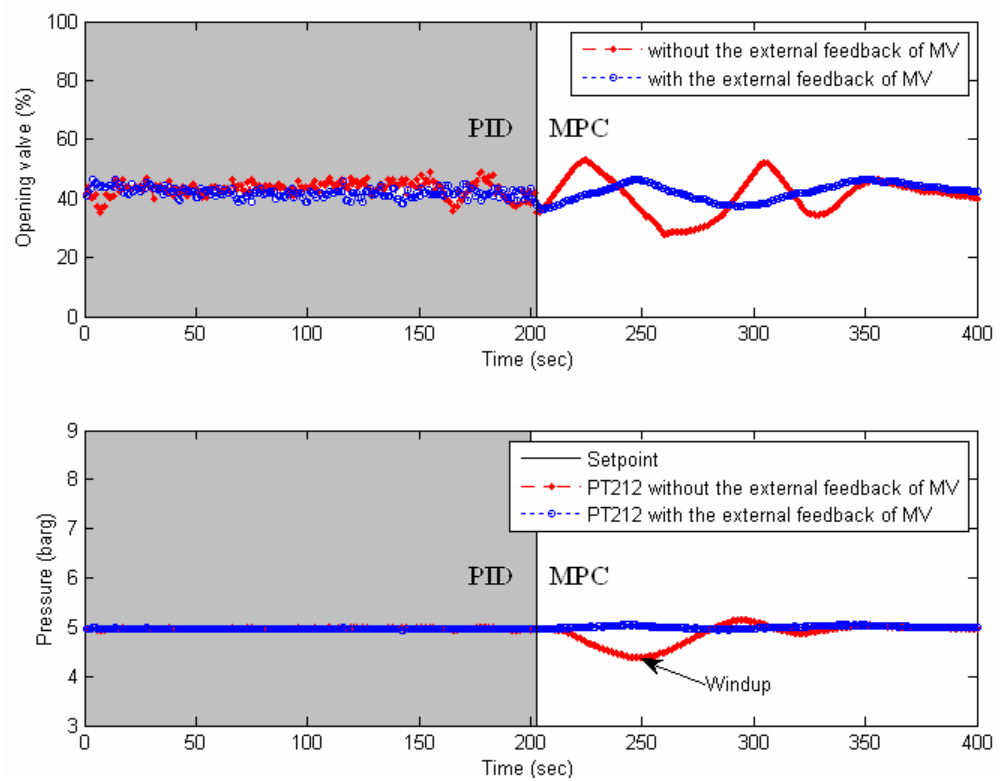
To produce the future control actions, the MPC controller needs to predict the future behaviors of the plant based on the internal model of the controller and the estimated current states of the plant. To estimate the current states, the state estimation equations in (4.6) to (4.8) are used. This requires the information about the past actions of plant inputs (manipulated variables) and plant outputs (process variables). Without any external feedback of manipulated variables, the only information about past actions of plant inputs is the signals that have been produced by MPC controller. The MPC controller assumes that the signals assigned to the actuators are equal with the outputs of MPC. Therefore, there will be error in the estimated states, and consequently the predictions of plant outputs for several steps ahead are no more valid. When the external feedback of manipulated variables is provided, the MPC controller will update the information about past actions of plant inputs, which are no longer equals to the past outputs of MPC controller. Hence, the estimation of the current states of plant remains valid.

Figure 4.32 shows the comparison of the manipulated variable and the process variable of upstream control system during switching time in the steady-state condition, from PID to MPC, with and without the external feedback of manipulated variables. As it can be seen, when the external feedback of manipulated variable is not provided, there is a sudden moves of manipulated variable just after the controller is switched from PID to MPC mode. As a consequence, during the transition time, the windup phenomenon will occur and the process variable will deviate from its setpoint. When the external feedback of manipulated variable is provided, the controller will make a smooth transition and the deviation of process variable from its setpoint (windup) disappears.

The comparison of the manipulated variable and the process variable of downstream control system during switching time in the steady-state condition, from PID to MPC, with and without the external feedback of manipulated variables, are shown in Figure 4.33. Inline with the response in upstream control system, the external feedback of manipulated variable provides the important information for the MPC controller to handle the transition during the switching time, thus provides the bumpless transfer of manipulated variable, and eliminates the deviation of process variable from its setpoint (windup).



**Figure 4.32:** The manipulated variable (top) and the process variable (bottom) of upstream control system during the switching time.



**Figure 4.33:** The manipulated variable (top) and the process variable (bottom) of downstream control system during the switching time.

#### 4.6.4 Discussion

As a continuation to the work on real-time implementation of model predictive control algorithm for an interacting series process, the study is extended to combining PID and MPC in the controller. The PID parameters are first determined using the Ziegler-Nichols closed-loop tuning method. The performances of PID controllers are then compared with the performance of MPC controller by examining their process variables and manipulated variables in the transient and steady-state conditions. It is found that the plant responses given by the MPC controller are slightly slower rather than the plant responses given by the PID controllers. However, the movements of manipulated variables of the PID controllers seem to give rapid correction of the errors, hence the extreme deviations of manipulated variables in the transient conditions occur. Furthermore, the manipulated variables of PID controllers are still presence in high frequencies during the steady-state conditions. In the real applications it may reduces the lifetime of the actuators. Since the objective of this study is not to do any comparisons to the performance of PID and MPC controllers, hence further investigation to improve the performance of PID control system in the gaseous pilot plant is not conducted.

This study is then narrowed down to the implementation of a simple method that able to make a smooth transition of manipulated variables during the switching from PID to MPC. The experiment results imply that the use of the external feedback of manipulated variables able to provide better estimation of the plant states and gives effective outcomes to make a bumpless transfer.

Based on the results, it can be concluded that it is possible to combine PID and MPC in the control system. The decentralized structure and the simplicity of PID will be needed when the process is in unusual conditions, such as for ease of start-up, automatic/manual transfer, and fault tolerance in the event of actuator or sensor failures. The multivariable control algorithm of MPC based on the minimization of its objective function may contribute a lot to achieve the economic and process benefits during the normal operating conditions.

#### 4.7 Summary

In this chapter, real-time implementation of model predictive control algorithm for controlling an interacting series process was presented. A real-time embedded control

system consisting of off-the-shelf components, such as standard PC and I/O modules, and software tools: MATLAB, Simulink, xPC Target, and RTW was implemented. Following the development and validation of the models discussed in the previous chapter, the developed models were tested as the internal model of MPC to examine their performances in real-time closed-loop control system. As the result, Model 3 showed to be robust against plant nonlinearities as indicated by zero steady-state errors. To improve its performances, the MPC controller was tuned by investigating the effect of four tuning parameters. As the outcomes, a simple set of guidelines to tune MPC was proposed. Considering several advantages of PID control system, a study of combining PID and MPC in the control system was performed. Detail analysis of the experiment results, it is possible to combine PID and MPC as a control system and there is a flexibility to switch them for their respective purposes.

## Chapter 5

### Conclusions and Future Work

#### 5.1 Conclusions

This thesis has discussed the empirical modeling using system identification technique and the implementation of the linear state-space model predictive control based on quadratic programming optimization method, with a focus on the interacting series processes. In general, a structure involving a series of systems occurs often in process control, and may also be combined with other structures, such as recycle structure or staged structure. The study has been carried out experimentally using a gaseous pilot plant as the process. The consideration was that the dynamic of the gaseous pilot plant exhibits the typical dynamic of an interacting series process, where the strong interaction between upstream and downstream properties occurs in both ways.

The empirical modeling was employed in the subspace system identification, which has been mentioned in many literatures to provide excellent technique for MIMO identification. To provide good estimations, two kinds of input signals, i.e., step and PRBS, were used, and three methods were taken into account for determining the model order. The challenge came when the nonlinearity occurs, while the identified model was expected to be robust against it. The main contribution is on to the modification of the model structure and performing MISO identifications, instead of MIMO identification. To assemble the models obtained using MISO identifications, the general formulation in state-space was derived. The open-loop model validation using a fresh data set, the APRBS response data, were performed during the first assessment of the identified models. Based on its structure, the model obtained from MISO identifications gave the best open-loop performance in term of the best fit criterions. Further model validation was carried out in real-time by utilizing the identified models as the internal model of a linear MPC controller. Yet, the state-space model obtained from MISO identifications showed its robustness, indicated by zero steady-state errors in the real-time implementation. Based on the results, taking into account the measurement of the upstream output as one of the inputs to predict the downstream output and vice versa, the structure of the model estimated using MISO identifications provides an effective prediction and control over the interacting series processes. Hence, the linear empirical



modeling of the interacting series process gives good robustness against plant nonlinearities.

The rapid prototyping of an embedded MPC controller has also been successfully developed using the available equipments, such as standard PC and I/O modules, and software package: MATLAB, Simulink, xPC Target, and Real-Time Workshop. Such experimental equipment was used to study the control of an interacting series process, the gaseous pilot, in real-time condition using MPC algorithm. The experiment results has demonstrated the applicability of the presented prototyping environment to implement the model predictive control algorithm in real-time. Moreover, there is no requirement on low level language programming.

The standard tuning rules of MPC have not been reported in any literatures, therefore, in this work the tuning of MPC controller was carried out in a heuristic way. Four parameters that could be used to adjust the performance of the MPC controller were examined. The four parameters are: the weights of MPC's optimization function, the prediction and control horizons, the state estimation gain, and the reference trajectories filters. As a result, the MPC controller was able to meet the desired performances: considerably fast to track the setpoints, no significant chattering or oscillation in steady-state conditions, and no overshoots. Based on the experiment results, a simple set of guidelines for tuning MPC controller has been proposed.

Considering several advantages of the decentralized control structure such as PID, and the fact that the linearized model is valid for certain neighborhood of operating points, the possibility of combining PID and MPC in the control system has also been studied. The applicability of the feedback of manipulated variables to make a bumpless transfer during the switching control from PID to MPC in steady-state condition has been illustrated.

## 5.2 Directions for Future Works

- **Input design for plant testing**

The PRBS input signals used in this thesis was designed on the assumption that the plant being modeled could linearly be approached with the first-order plus delay time transfer functions of each output from their respective inputs. The plant also perturbed simultaneously with three different PRBS signals without any further design. The fact

that this setting gave poor estimation for MIMO identification may give an insight for further investigation on how the PBRS signals should be designed and as an alternative on the open-loop perturbation.

- **Further alternatives on the use of the estimated model**

Further applications of the estimated model, such as for soft sensing or dynamic data reconciliation applications could be beneficial to be considered, examined, and experimentally tested.

- **Computational capacity of the rapid prototyping environment**

The long prediction horizon of MPC controller may serve better control actions. However, increasing the horizons of MPC increases the computational load of the controller. The excessive setting of the horizons may cause the optimization function unable to be solved by the controller during every sampling time. The investigation on the computational capacity of the rapid prototyping environment being used may necessarily to be conducted so that the memory and the processing tasks of the controller in every sampling time can be allocated properly.

- **Further advances in system identification and model predictive control**

The further advances in system identification such as closed-loop identification or nonlinear identification could be very interesting to be investigated. The use of MPC algorithm as a dynamic optimizer of the plant to make a good coordination between low-level control loops may also worthwhile to be examined.

The work presented in this thesis has contributed to an improved understanding of the procedure for modeling, analysis, system identification, and model predictive control development of an interacting series process. Even though the approach presented here offers some promising insights, much work remains to be done to improve the technique under different types of systems to demonstrate the generality of the approach. Several directions of future work presented above might serve an insight on the further investigations related to this work.

## References

- [1] A. Faanes, "Controllability analysis for process and control system design," Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2003.
- [2] E. M. B. Aske, "Design of plantwide control systems with focus on maximizing throughput," Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2009.
- [3] S. Skogestad, "Control structure design for complete chemical plants," *Computers & Chemical Engineering*, vol. 28, pp. 219-234, 2004.
- [4] J. Morud, "Studies on the dynamics and operation of integrated plants," Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway, 1996.
- [5] F. Xu, B. Huang, and S. Akande, "Performance assessment of model predictive control for variability and constraint tuning," *Ind. Eng. Chem. Res.*, vol. 46, pp. 1208-1219, 2007.
- [6] J. Gao, R. Patwardhana, K. Akamatsub, Y. Hashimoto, G. Emoto, S. L. Shah, and B. Huang, "Performance evaluation of two industrial MPC controllers," *Control Engineering Practice*, vol. 11, pp. 1371-1387, 2003.
- [7] N. Agarwal and B. Huang, "Assessing model prediction control (MPC) performance. 1. Probabilistic approach for constraint analysis," *Ind. Eng. Chem. Res.*, vol. 46, pp. 8101-8111, 2007.
- [8] D. Weia, I. K. Craig, and M. Bauer, "Multivariate economic performance assessment of an MPC controlled electric arc furnace," *ISA Transactions*, vol. 46, pp. 429-436, 2007.
- [9] M. Bauer and I. K. Craig, "Economic assessment of advanced process control – A survey and framework," *Journal of Process Control*, vol. 18, pp. 2-18, 2008.
- [10] J. Z. Lu, "Challenging control problems and emerging technologies in enterprise optimization," *Control Engineering Practice*, vol. 11, pp. 847-858, 2003.

- [11] A. Al-Ghazzawi and B. Lennox, "Model predictive control monitoring using multivariate statistics," *Journal of Process Control*, vol. 19, pp. 314-327, 2009.
- [12] K. H. Lee, B. Huang, and E. C. Tamayo, "Sensitivity analysis for selective constraint and variability tuning in performance assessment of industrial MPC," *Control Engineering Practice*, vol. 16, pp. 1195-1215, 2008.
- [13] V. Adetola, D. DeHaan, and M. Guay, "Adaptive model predictive control for constrained nonlinear systems," *Systems & Control Letters*, vol. 58, pp. 320-326, 2009.
- [14] F. A. Cuzzola, J. C. Geromel, and M. Morari, "An improved approach for constrained robust model predictive control," *Automatica*, vol. 38, pp. 1183-1189, 2002.
- [15] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Transactions on Automatic Control*, vol. 45, pp. 1545-1549, 2000.
- [16] M. A. Rodrigues and D. Odloak, "MPC for stable linear systems with model uncertainty," *Automatica*, vol. 39, pp. 569-583, 2003.
- [17] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, pp. 219-224, 2005.
- [18] S. B. Jørgensen and J. H. Lee, "Recent advances and challenges in process identification," *AIChE Symposium Series*, vol. 98, pp. 55-74, 2002.
- [19] Y. Zhu and P. Stec, "Simple control-relevant identification test methods for a class of ill-conditioned processes," *Journal of Process Control*, vol. 16, pp. 1113-1120, 2006.
- [20] B. S. Dayal and J. F. MacGregor, "Multi-output process identification," *Journal of Process Control*, vol. 7, pp. 269-282, 1997.
- [21] G. R. Srinivas and Y. Arkun, "Control of the Tennessee Eastman process using input-output models," *Journal of Process Control*, vol. 7, pp. 387-400, 1997.
- [22] B. C. Juriceka, D. E. Seborg, and W. E. Larimore, "Identification of the Tennessee Eastman challenge process with subspace methods," *Control Engineering Practice*, vol. 9, pp. 1337-1351, 2001.

- [23] L. G. Bleris and M. V. Kothare, "Real-time implementation of model predictive control," in *2005 American Control Conference*, Portland, USA, 2005.
- [24] A. Bemporad, "Model predictive control design: New trends and tools," in *45th IEEE Conference on Decision & Control*, San Diego, CA, USA, 2006.
- [25] P. Dua, K. Kouramas, V. Dua, and E. N. Pistikopoulos, "MPC on a chip - Recent advances on the application of multi-parametric model-based control," *Computers & Chemical Engineering*, vol. 32, pp. 754-765, 2008.
- [26] R. Shridar and D. J. Cooper, "A novel tuning strategy for multivariable model predictive control," *ISA Transactions*, vol. 36, pp. 273-280, 1998.
- [27] A. Al-Ghazzawi, E. Ali, A. Nouh, and E. Zafiriou, "On-line tuning strategy for model predictive controllers," *Journal of Process Control*, vol. 11, pp. 265-284, 2001.
- [28] J. O. Trierweilera and L. A. Farina, "RPN tuning strategy for model predictive control," *Journal of Process Control*, vol. 13, pp. 591-598, 2003.
- [29] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta, "Practical approach to tuning MPC," *ISA Transactions*, vol. 42, pp. 149-162, 2003.
- [30] L. O. Santos, L. T. Biegler, and J. A. A. M. Castro, "A tool to analyze robust stability for constrained nonlinear MPC," *Journal of Process Control*, vol. 18, pp. 383-390, 2008.
- [31] L. O. Santos, P. A. F. N. A. Afonso, J. A. A. M. Castro, N. M. C. Oliveira, and L. T. Biegler, "On-line implementation of nonlinear MPC: An experimental case study," *Control Engineering Practice*, vol. 9, pp. 847-857, 2001.
- [32] V. M. Zavala and L. T. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, pp. 86-93, 2009.
- [33] R. Romana, Z. K. Nagy, M. V. Cristea, and S. P. Agachi, "Dynamic modelling and nonlinear model predictive control of a Fluid Catalytic Cracking Unit," *Computers & Chemical Engineering*, vol. 33, pp. 605-617, 2009.
- [34] R. Dubay, M. Abu-Ayyad, and J. M. Hernandez, "A nonlinear regression model-based predictive control algorithm," *ISA Transactions*, vol. 48, pp. 180-189, 2009.

- [35] T. Li and C. Georgakis, "Dynamic input signal design for the identification of constrained systems," *Journal of Process Control*, vol. 18, pp. 332-346, 2008.
- [36] A. Micchi and G. Pannocchia, "Comparison of input signals in subspace identification of multivariable ill-conditioned systems," *Journal of Process Control*, vol. 18, pp. 582-593, 2008.
- [37] A. Bemporad, M. Morari, and N. L. Ricker, *Model Predictive Control Toolbox™ 2 User's Guide*. Natick, MA: The MathWorks, Inc., 2008.
- [38] B. Roffel and B. Betlem, *Process Dynamics and Control: Modeling for Control and Prediction*. West Sussex: John Wiley & Sons Ltd, 2006, pp. 25-55.
- [39] J. Morud and S. Skogestad, "Dynamic behaviour of integrated plants," *Journal of Process Control*, vol. 6, pp. 145-156, 1996.
- [40] T. E. Marlin, *Process Control: Designing Processes and Control Systems for Dynamic Performance*. Singapore: McGraw-Hill Companies Inc., 2000, pp. 143-166, 239-350, 583-757.
- [41] A. Faanes and S. Skogestad, "Controller design for serial processes," *Journal of Process Control*, vol. 15, pp. 259-271, 2005.
- [42] P. Albertos and A. Sala, *Multivariable Control System: An Engineering Approach*. London: Springer-Verlag, 2004, pp. 1-16.
- [43] T. Katayama, *Subspace Methods for System Identification*. London: Springer, 2005, pp. 1-13.
- [44] L. Ljung, *System Identification: Theory for The User*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, Inc., 1999, pp. 1-17.
- [45] P. V. Overschee and B. D. Moor, *Subspace Identification for Linear Systems*. London: Kluwer Academic Publishers, 1996, pp. 1-12.
- [46] S. D. Fassois and D. E. Rivera, "Applications of system identification," *IEEE Control Systems Magazine*, vol. 27, pp. 24-26, October 2007.
- [47] T. Bohlin, *Practical Grey-Box Process Identification : Theory and Applications*. London: Springer-Verlag Ltd., 2006, pp. 1-22.
- [48] L. Jin, R. Kumar, and N. Elia, "Application of model predictive control in voltage stabilization," in *2007 American Control Conference*, New York City, USA, 2007.

- [49] H. Zabiri and Y. Samyudia, "A hybrid formulation and design of model predictive control for system under actuator saturation and backlash," *Journal of Process Control*, vol. 16, pp. 693-709, 2006.
- [50] M. Morari and M. Barić, "Recent developments in the control of constrained hybrid system," *Computers & Chemical Engineering*, vol. 30, pp. 1619-1631, 2006.
- [51] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733-764, 2003.
- [52] B. Likar and J. Kocijan, "Predictive control of gas-liquid separation plant based on a Gaussian process model," *Computers & Chemical Engineering*, vol. 31, pp. 142-152, 2007.
- [53] C. Bordons and E. F. Camacho, "A generalized predictive controller for a wide class of industrial processes," *IEEE Transactions on Control Systems Technology*, vol. 6, pp. 372-387, 1998.
- [54] J. B. Froisy, "Model predictive control - building a bridge between theory and practice," *Computers & Chemical Engineering*, vol. 30, pp. 1426-1435, 2006.
- [55] Y. Majanne, "Model predictive pressure control of steam networks," *Control Engineering Practice*, vol. 13, pp. 1499-1505, 2005.
- [56] C. R. Cuttler and B. L. Ramaker, "Dynamix matrix control - a computer control algorithm," in *Joint Automatic Control Conference*, San Fransisco, California, USA, 1980.
- [57] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, pp. 413-428, 1978.
- [58] R. K. Mehra, R. Rouhani, J. Eterno, J. Richalet, and A. Rault, "Model algorithmic control: Review and recent developments," in *Engineering Foundation Conference (Chemical Process Control II)*, Georgia, USA, 1981.
- [59] J. B. Froisy and J. Richalet, "Industrial applications of IDCOM," in *Third International Conference on Chemical Process Control*, Amsterdam, The Netherlands, 1986.

- [60] C. E. Garcia and D. M. Prett, "Advances in industrial model-predictive control," in *Third International Conference on Chemical Process Control*, Amsterdam, The Netherlands, 1986.
- [61] M. Morari and J. L. Lee, "Model predictive control: The good, the bad and the ugly," in *Fourth International Conference on Chemical Process Control*, Amsterdam, The Netherlands, 1991.
- [62] N. L. Ricker, "Model predictive control: State of the art," in *Fourth International Conference on Chemical Process Control*, Amsterdam, The Netherlands, 1991.
- [63] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," in *Fifth International Conference on Chemical Process Control*, Tahoe City, CA, USA, 1996.
- [64] K. R. Muske and J. B. Rawlings, "Model predictive control with linear models," *AIChE Journal*, vol. 39, pp. 262-287, 1993.
- [65] S. J. Wright, "Applying new optimization algorithms to model predictive control," in *Fifth International Conference on Chemical Process Control*, Tahoe City, CA, USA, 1996.
- [66] E. F. Vogel and J. J. Downs, "Industrial experience with state-space model predictive control," in *Sixth International Conference on Chemical Process Control*, Tucson, Arizona, USA, 2001.
- [67] R. E. Young, R. D. Bartusiak, and R. W. Fontaine, "Evolution of an industrial nonlinear predictive controller," in *Sixth International Conference on Chemical Process Control*, Tucson, Arizona, USA, 2001.
- [68] E. F. Camacho and C. Bordons, *Model Predictive Control*. London: Springer, 2004, pp. 1-192.
- [69] J. M. Maciejowski, *Predictive Control with Constraints*. London: Prentice Hall, 2002, pp. 36-104.
- [70] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control for constrained linear systems under uncertainty based on feed forward and positive invariant feedback control," in *IEEE Conference on Decision & Control*, San Diego, CA, USA, 2006.



- [71] L. L. Giovanini, "Model predictive control with amplitude and rate actuator saturation," *ISA Transactions*, vol. 42, pp. 227-240, 2003.
- [72] L. Feng, J. L. Wang, and E. K. Poh, "Computational complexity reduction for robust model predictive control," in *International Conference on Control and Automation*, Budapest, Hungary, 2005.
- [73] T. J. J. v. d. Booma, B. Heidergott, and B. D. Schuttera, "Complexity reduction in MPC for stochastic max-plus-linear discrete event systems by variability expansion," *Automatica*, vol. 43, pp. 1058-1063, 2007.
- [74] R. Bălan, V. Mătieș, O. Hancu, and S. Stan, "A predictive control approach for the inverse pendulum on a cart problem," in *IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, 2005.
- [75] O. A. Z. Sotomayor, S. W. Parka, and C. Garcia, "Multivariable identification of an activated sludge process with subspace-based algorithms," *Control Engineering Practice*, vol. 11, pp. 961-969, 2003.
- [76] P. Stoica and M. Jansson, "MIMO system identification: State-space and subspace approximations versus transfer function and instrumental variables," *IEEE Transactions on Signal Processing*, vol. 48, pp. 3087-3099, 2000.
- [77] A. Faanes and S. Skogestad, "pH-neutralization: Integrated process and control design," *Computers & Chemical Engineering*, vol. 28, pp. 1475-1487, 2004.
- [78] E. P. Gatzke, E. S. Meadows, C. Wang, and F. J. Doyle, "Model based control of a four-tank system," *Computers & Chemical Engineering*, vol. 24, pp. 1503-1509, 2000.
- [79] E. Weyer, "System identification of an open water channel," *Control Engineering Practice*, vol. 9, pp. 1289-1299, 2001.
- [80] T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "Multivariable system identification of gaseous pilot plant," in *National Postgraduate Conference on Engineering, Science, and Technology*, Tronoh, Perak, Malaysia, 2008.
- [81] T. Bastogne, H. Noura, P. Sibille, and A. Richard, "Multivariable identification of a winding process by subspace methods for tension control," *Control Engineering Practice*, vol. 6, pp. 1077-1088, 1998.

- [82] W. Favoreel, B. D. Moor, and P. V. Overschee, "Subspace state space system identification for industrial processes," *Journal of Process Control*, vol. 10, pp. 149-155, 2000.
- [83] B. Lee, Y. Kim, D. Shin, and E. S. Yoon, "A study on the evaluation of structural controllability in chemical processes," *Computers & Chemical Engineering*, vol. 25, pp. 85–95, 2001.
- [84] D. Bauer, "Order estimation for subspace methods," *Automatica*, vol. 37, pp. 1561-1573, 2001.
- [85] Y. Zhang, Z. Zhang, X. Xu, and H. Hua, "Modal parameter identification using response data only," *Journal of Sound and Vibration*, vol. 282, pp. 367-380, 2005.
- [86] L. Mevel, A. Benveniste, M. Basseville, and M. Goursat, "Blind subspace-based eigenstructure identification under nonstationary excitation using moving sensors," *IEEE Transactions on Signal Processing*, vol. 50, pp. 41-48, 2002.
- [87] G. Balas, R. Chiang, A. Packard, and M. Safonov, *Robust Control Toolbox™ 3 Getting Started Guide*. Natick, MA: The MathWorks, Inc., 2008.
- [88] J. Hahn and T. F. Edgar, "An improved method for nonlinear model reduction using balancing of empirical gramians," *Computers & Chemical Engineering*, vol. 26, pp. 1379-1397, 2002.
- [89] M. G. Safonov and R. Y. Chiang, "A Schur method for balanced-truncation model reduction," *IEEE Transactions on Automatic Control*, vol. 34, pp. 729-733, 1989.
- [90] A. C. Antoulas, D. C. Sorensen, and Y. Zhou, "On the decay rate of Hankel singular values and related issues," *Systems & Control Letters*, vol. 46, pp. 323-342, 2002.
- [91] L. Ljung, *System Identification Toolbox™ 7 User's Guide*. Natick, MA: The MathWorks, Inc., 2008.
- [92] C. Brosilow and B. Joseph, *Techniques of Model-Based Control*. Upper Saddle River, New Jersey: Prentice Hall PTR, 2002, pp. 387-393.
- [93] D. E. Rivera and S. V. Gaikwad, "Systematic techniques for determining modeling requirements for SISO and MIMO feedback control," *Journal of Process Control*, vol. 4, pp. 213-224, 1995.

- [94] M. Albaz, S. Karacan, Y. Cabbar, and H. Hapoğlu, "Application of model predictive control and dynamic analysis to a pilot distillation column and experimental verification," *Chemical Engineering Journal*, vol. 88, pp. 163-174, 2002.
- [95] F. Wu, "LMI-based robust model predictive control and its application to an industrial CSTR problem," *Journal of Process Control*, vol. 11, pp. 649-659, 2001.
- [96] C. R. Porfirio, E. A. Neto, and D. Odloak, "Multi-model predictive control of an industrial C3/C4 splitter," *Control Engineering Practice*, vol. 11, pp. 765-779, 2003.
- [97] V. Havlena and J. Findejs, "Application of model predictive control to advanced combustion control," *Control Engineering Practice*, vol. 13, pp. 671-680, 2005.
- [98] H. Huang and J. B. Riggs, "Comparison of PI and MPC for control of a gas recovery unit," *Journal of Process Control*, vol. 12, pp. 163-173, 2002.
- [99] A. H. González, D. Odloak, and J. L. Marchetti, "Predictive control applied to heat-exchanger networks," *Chemical Engineering and Processing*, vol. 45, pp. 661-671, 2006.
- [100] Y. Zhanga and S. Li, "Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes," *Journal of Process Control*, vol. 17, pp. 37-50, 2007.
- [101] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 559-576, 2005.
- [102] D. Pomerleau, A. Pomerleau, D. Hodouin, and É. Poulin, "A procedure for the design and evaluation of decentralised and model-based predictive multivariable controllers for a pellet cooling process," *Computers & Chemical Engineering*, vol. 27, pp. 217-233, 2003.
- [103] P. Nordfeldt and T. Hägglund, "Decoupler and PID controller design of TITO systems," *Journal of Process Control*, vol. 16, pp. 923-936, 2006.

- [104] V. Puig and J. Quevedo, "Fault-tolerant PID controllers using a passive robust fault diagnosis approach," *Control Engineering Practice*, vol. 9, pp. 1221-1234, 2001.
- [105] J. Bao, W. Z. Zhang, and P. L. Lee, "Decentralized fault-tolerant control system design for unstable processes," *Chemical Engineering Science*, vol. 58, pp. 5045-5054, 2003.
- [106] S. Yuan, "A study of model predictive control on variable-air-volume air conditioning systems," Ph.D. thesis, The University of Wisconsin-Milwaukee, Wisconsin, USA, 2007.
- [107] X. Zhang and K. A. Hoo, "Control of an integrated biological wastewater treatment system," in *16th IEEE International Conference on Control Applications*, Singapore, 2007.
- [108] E. Kreyszig, *Advanced Engineering Mathematics*, 9th ed. Singapore: John Wiley & Sons, Inc., 2006, pp. 349-356.
- [109] R. Fletcher, *Practical Methods of Optimization*. Great Britain: John Wiley & Sons Ltd., 2000, pp. 229-255.
- [110] G. C. Goodwin, M. M. Seron, and J. A. D. Dona, *Constrained Control and Estimation: An Optimisation Approach*. London: Springer, 2005, pp. 131-133.
- [111] K. V. Ling, S. P. Yue, and J. M. Maciejowski, "A FPGA Implementation of Model Predictive Control," in *2006 American Control Conference*, Minneapolis, Minnesota, USA, 2006.
- [112] D. Hristu-Varsakelis and W. S. Levine, *Handbook of Networked and Embedded Control Systems*. Boston: Birkhäuser, 2005, pp. 419-446.
- [113] P. S. Shiakolas and D. Piyabongkarn, "On the development of a real-time digital control system using xPC-Target and a magnetic levitation device," in *40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, 2001.
- [114] -, *xPC Target User's Guide*. Natick, MA: The MathWorks Inc., 2004.
- [115] K. H. Low, H. Wang, and M. Y. Wang, "On the development of a real time control system by using xPC Target: Solution to robotic system control," in *IEEE*

*International Conference on Automation Science and Engineering*, Edmonton, Canada, 2005.

- [116] F. Bezzo, F. Micheletti, R. Muradore, and M. Barolo, "Using MPC to control middle-vessel continuous distillation columns," *Journal of Process Control*, vol. 15, pp. 925-930, 2005.
- [117] M. S. Elliott and B. P. Rasmussen, "Model-based predictive control of a multi-evaporator vapor compression cooling cycle," in *2008 American Control Conference*, Seattle, Washington, USA, 2008.
- [118] J. D. Ward and C.-C. Yu, "Population balance modeling in Simulink: PCSS," *Computers & Chemical Engineering*, vol. 32, pp. 2233-2242, 2008.
- [119] C. Muñoz, D. Rojas, O. Candia, L. Azocar, C. Bornhardt, and C. Antileo, "Supervisory control system to enhance partial nitrification in an activated sludge reactor," *Chemical Engineering Journal*, vol. 145, pp. 453-460, 2009.
- [120] M.-J. Bruwer and J. F. MacGregor, "Robust multi-variable identification: Optimal experimental design with constraints," *Journal of Process Control*, vol. 16, pp. 581-600, 2006.
- [121] J. Schäfer and A. Cinar, "Multivariable MPC system performance assessment, monitoring, and diagnosis," *Journal of Process Control*, vol. 14, pp. 113-129, 2004.
- [122] T. C. S. Wibowo, N. Saad, and M. N. Karsiti, "The simulation of MISO MPC for gaseous pilot plant control with presence of measurement noise," in *International Conference on Intelligent and Advanced Systems*, Kuala Lumpur, Malaysia, 2007.
- [123] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," in *Fifth International Conference on Chemical Process Control*, 1997.
- [124] S. J. Qin, "Control performance monitoring -- a review and assessment," *Computers & Chemical Engineering*, vol. 23, pp. 173-186, 1998.
- [125] Y. Peng, D. Vrancic, and R. Hanus, "Anti-windup, bumpless, and conditioned transfer techniques for PID controllers," *IEEE Control Systems Magazine*, vol. 16, pp. 48-57, August 1996.

- [126] L. Zaccarian and A. R. Teel, "A common framework for anti-windup, bumpless transfer and reliable designs," *Automatica*, vol. 38, pp. 1735-1744, 2002.
- [127] A. S. Hodel and C. E. Hall, "Variable-structure PID control to prevent integrator windup," *IEEE Transactions on Industrial Electronics*, vol. 48, pp. 442-451, 2001.
- [128] Y. Li, K. H. Ang, and G. C. Y. Chong, "PID control system analysis and design," *IEEE Control Systems Magazine*, vol. 26, pp. 32-41, 2006.
- [129] K. Zheng, A. H. Lee, J. Bentsman, and C. W. Taft, "Steady-state bumpless transfer under controller uncertainty using the state/output feedback topology," *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 3-17, 2006.
- [130] A. Bemporad, A. R. Teel, and L. Zaccarian, "Anti-windup synthesis via sampled-data piecewise affine optimal control," *Automatica*, vol. 40, pp. 549-562, 2004.
- [131] E. F. Mulder, M. V. Kothare, and M. Morari, "Multivariable anti-windup controller synthesis using linear matrix inequalities," *Automatica*, vol. 37, pp. 1407-1416, 2001.
- [132] S. Galeani, A. R. Teel, and L. Zaccarian, "Constructive nonlinear anti-windup design for exponentially unstable linear plants," *Systems & Control Letters*, vol. 56, pp. 357-365, 2007.
- [133] W. L. Luyben, *Process Modeling, Simulation, and Control for Chemical Engineers*. Singapore: McGraw-Hill Inc., 1999, pp. 235-237.